Statistics and Algorithms for Peptide Mass Fingerprinting

Dipl.-Math. Hans-Michael Kaltenbach

Thesis submitted to the
Faculty of Technology, Bielefeld University, Germany
for the degree of Dr. rer. nat.

Date of thesis defense: **20. March 2007**

Supervisors
Prof. Dr. Sebastian Böcker
Dr. Sven Rahmann

Referees
Prof. Dr. Sebastian Böcker, Jena University
Dr. Sven Rahmann, Bielefeld University

Gedruckt auf alterungsbeständigem Papier nach DIN-ISO 9706
(Printed on non-aging paper according to DIN-ISO 9706)

Abstract

We investigate several mathematical and algorithmical aspects of peptide mass fingerprinting (PMF). In a PMF experiment, a purified protein sample is digested by a protease using an enzymatic cleavage reaction, the masses of the resulting peptides are measured by mass spectrometry, yielding the mass fingerprint, and compared to predicted mass fingerprints of reference protein sequences.

In the first part, we examine several statistics of PMFs. We introduce random weighted strings over *probabilistically weighted alphabets* and *cleavage schemes* as mathematical models for random protein sequences, their molecular mass, and their mass fingerprints. We examine *weighted hidden Markov models* and *Markov additive chains* as a general computational framework for the stochastics of protein fragments and their masses. In parallel, we present recurrence equations for the description of these fragments.

Using these models, the distribution of fragment lengths, the distribution of fragment masses and the distribution of the number of fragments in a random protein are examined under a random string model of independent, identically distributed characters. We derive the *occurrence probability* of a certain fragment mass in a random protein sequence of either fixed length or fixed mass.

We present efficient dynamic programming algorithms and their time and space complexity for most of the statistics and compare all statistics with their empirical counterparts estimated from an in-silico tryptic digest of the Swiss-Prot protein sequence database.

In the second part, we develop a general algorithmic framework for identification of PMFs with a protein sequence database search. We formalize the identification of a mass spectrum as an alignment problem and modify well-known methods developed for sequence analysis to an efficient algorithm for computing an optimal alignment of a measured spectrum and a predicted spectrum. The alignment is based on *scoring schemes* that allow flexible and consistent incorporation of a multitude of experimental parameters such as accuracy of the measured masses, mass error distribution, sample contamination, and ionization efficiency into the identification procedure.

Using the fragment statistics of the first part, we estimate the significance of a protein identification under a well-defined statistical null-model.

Finally, we present a family of scoring schemes and show how additional information such as intensity values of measured peaks can be incorporated into an alignment scoring. We demonstrate the applicability of the alignment framework on a real proteomics dataset and compare our results with the standard software MASCOT.

Contents

1. Proteomics – Biological Background

According to Palagi *et al* [102], the word *"proteome"* was introduced in 1994 to denote the protein complement of the genome. It refers to the complete set of proteins present in a cell at a certain time. Unlike the genome, a proteome is dynamic: Proteins are constantly built and degraded and their presence and abundances depend on a multitude of factors like cell type, growth state, or external stress. One of the major aspects of *proteomics* – the study of the proteome – is the identification of all proteins present in a cell or tissue at a certain time.

Protein structure. Proteins are involved in almost all cellular activities: They are part of the cell membrane, work as pumps for ion exchange, serve as receptors and transmitters in signal transduction, and are involved in metabolic networks as enzymes, catalysts and inhibitors.

Proteins are *polypeptides*: They are built as a chain consisting of several peptides which in turn are polymers formed by a chain of *amino acid molecules*. Thus, proteins are long polymers of amino acids. The terminology is not very precise: Usually, a polypeptide that has some biological function is called a protein (cf. [85]).

Although each amino acid has unique chemical and physical properties such as hydrophobicity or polarity, all amino acids share the same common structure. A central carbon atom is surrounded by an amine group (NH_2), a carboxyl group (COOH), a hydrogen atom and a residue (usually denoted R) specific for the amino acid. On the left side of Figure 1.1, two amino acids with residues R_1, R_2 are depicted.

The three-letter code and the one-letter code are two equivalent notations for the 20 different amino acids most commonly used in organisms. Table 1.1 lists these 20 amino acids together with their codes, monoisotopic and average mass (see below), structural formula and frequency of occurrence in protein sequences contained in the Swiss-Prot database [10]. Masses are given in Dalton (Da), where one Dalton is approximately the mass of a single proton.

A polymer of amino acids is formed by building *peptide bonds* between two amino acids, involving the separation of a water molecule, see Figure 1.1 for an illustration. Each peptide has an unbounded carboxyl group, called the *C-terminus*, and an unbounded amine group, called the *N-terminus*. The amino acid molecules within a peptide, without the water molecule lost by forming a peptide bond, are called the amino acid residues of the peptide. If a peptide bond is broken, a water molecule is attached to the two new terminal residue amino acids, completing the new N-terminus and C-terminus, respectively.

The sequence of amino acids, read from N- to C-terminus, is called the *primary structure* of the protein; it uniquely determines the protein. Certain peptides of the protein fold into characteristic shapes, either into a *α-helix*, a *β-sheet* or a *random coil*. These

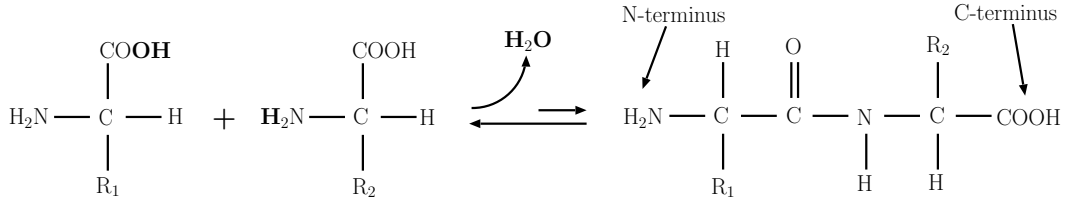

Figure 1.1.: The peptide bond between two amino acids with residues R_1, R_2.

shapes are called the *secondary structure* of the protein. However, the function of a protein is mostly determined by its three-dimensional structure, also called the *conformation* of a protein or its *tertiary structure*. Proteins fold into their conformation; it usually provides the energetic minimum of the molecule. The *native conformation* is determined by the primary structure. However, the conformation is changed if certain chemical groups, such as a phosphor molecule, are attached to the protein.

Protein synthesis. The genome consists of several *deoxyribonucleic acid (DNA)* molecules, each a double-stranded polymer build from 4 different *nucleic acids*: *adenine* (one-letter code A), *cytosine* (C), *guanine* (G) and *thymine* (T). The genome is static and the same for all cells of an organism. The primary structure of a protein is encoded in a *gene*, a *coding region* of the genome. However, some genes also encode special RNA molecules (see below), so there is no one-to-one correspondence of genes and proteins.

If the synthesis of a new protein is triggered by cellular signals, the gene encoding this protein is copied from the DNA to a *messenger ribonucleic acid (mRNA)* molecule. RNA molecules are almost like DNA molecules, except that they are single-stranded, use a different ribose in the backbone and use *uracil* (U) instead of thymine (T).

In eucaryotes, parts of the mRNA sequence, so-called *introns*, are cleaved out; this does not happen in procaryotes. This *splicing process* is not unique, there usually exist several possibilities which parts are cleaved, leading to *alternative splicings*, and different proteins are encoded by the same gene. The mRNA is also altered by appending certain signaling sequences, such as the poly-A-tail (eucaryotes only), for guiding the transfer of the molecules to the ribosomes (see below) and for controlling the process of translation to protein. The whole process of copying genetic information and building the mRNA molecule including the different alterations is called *transcription*. Similar to the genome and the proteome, the *transcriptome* refers to the set of all mRNA molecules present in a cell at a certain time. Just like the proteome, the transcriptome is dynamic.

The mRNA molecules are transferred to the *ribosomes*, large ribonucleoproteins that are responsible for building the protein's primary structure from amino acids. The process of *translation* of the mRNA sequence to an amino acid sequence is done with the help of transfer RNA (tRNA) that transfers the amino acids to the amino acid chain. See Figure 1.2 for a schematic illustration of transcription and translation.

A triplet of nucleic acids encodes one amino acid by the genetic code (cf. Table 1.2); the triplets are also called *codons*. Codons are consecutive and non-overlapping, i.e., the next three nucleotides after each codon form the next codon. The start and end of

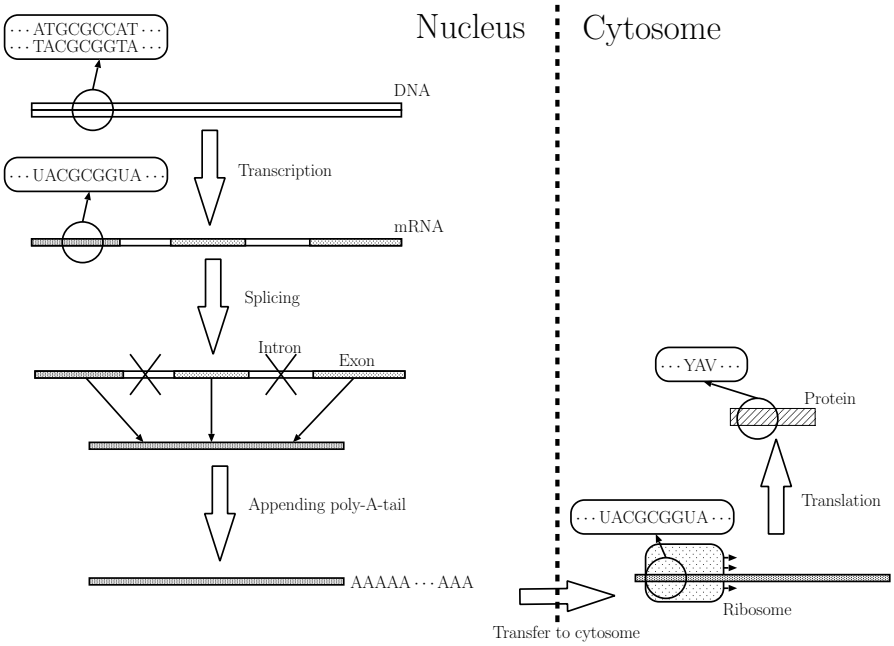

Figure 1.2.: Schematic process of transcription and translation in eucaryotes.

the mRNA sequence translation process is triggered by special *start and stop codons*. The start codon AUG encodes the amino acid methionine. In almost all organisms, the newly synthesized protein thus starts with a methionine; it is usually cleaved from the protein right after synthesis and is thus not part of the primary structure. The three stop codons do not encode any amino acid.

The Central Dogma. The *Central Dogma of molecular biology* states that the flow of information in a cell is always from DNA to RNA to protein (cf. Figure 1.3). The genetic information of a cell is copied by replication of the DNA molecules. It can be transcribed into RNA molecules, but usually, information encoded in RNA cannot be transfered into DNA; there is the exception of retro-viruses that actually transfer their genetic information encoded in RNA into the host cells' DNA by reverse transcriptase. Information encoded in RNA can be translated into proteins, but proteins can alter neither RNA nor DNA. In particular, if the genome of an organism is sequenced, i.e. its DNA sequence is revealed, and the coding regions and genes are identified within the sequences, all possible protein sequences that can be translated from the genome are in principle also known. With a growing number of genomes being sequenced, protein sequence databases are built by in-silico prediction of genes and their protein product from the DNA sequences. Recall that unlike the genome, the proteome is not static and it is thus not sufficient to know all possible protein sequences. However, the hypothetical protein sequences are a major tool for identification of observed proteins by searching sequence databases using mass spectrometric data, see Chapter 2.

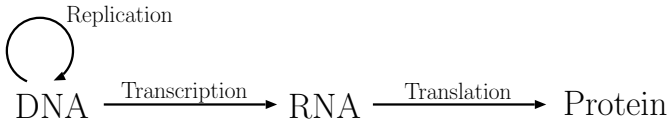

Figure 1.3.: The flow of information in a cell according to the Central Dogma.

Post-translational modifications. Besides the question of absence or presence of a particular protein, there is another problem that aggravates the analysis of the proteome: Proteins are also altered after translation by *post-translational modifications (PTMs)*. PTMs are usually attachments of certain chemical groups to one or more of the amino acids of a protein; they can be specific for certain amino acids. Some of the most common PTMs are *glycosylations* and *phosphorylations*.

Glycosylations play an important role in the immunology of higher organisms. Proteins on the outer cell membrane use the attached sugars to distinguish between body and non-body cells and to recognize messengers like hormones and neurotransmitters on receptors. The set of all glycosylations of membrane proteins is called the *glycocalix* of the cell; it distinguished blood cells of type A and B, for example.

A typical example for phosphorylation of a protein is the regulation of ion channel proteins that do an active transport of ions like Ca^{2+} through the cell membrane. These proteins are usually "activated" by attaching a phosphor molecule that causes a change in the protein's three-dimensional conformation and thus opens or closes an ion channel. The phosphor typically originates from a transition of ATP (adenosine-tri-phosphate) to ADP (adenosine-di-phosphate), an energy system common to all cells.

Proteins of the same primary structure but different modifications are called *isoforms*. It is assumed that the $\sim$ 35 000 genes of the human genome encode up to 100 000 functional proteins and there might exist up to 1 000 000 different possible functional isoforms in the human body [7].

amino acid	3-code	1-code	monoiso. mass (Da)	avg mass (Da)	mol. composition	freq. (%)
Alanine	Ala	A	71.037113790	71.079323045	$C_3H_5N_1O_1$	7.85
Arginine	Arg	R	156.101111044	156.188746822	$C_6H_{12}N_4O_1$	5.33
Asparagine	Asn	N	114.042927452	114.104467719	$C_4H_6N_2O_2$	4.18
Aspartatic acid	Asp	D	115.026943030	115.089069711	$C_4H_5N_1O_3$	5.31
Cysteine	Cys	C	103.009184490	103.143711176	$C_3H_5N_1O_1S_1$	1.54
Glutamic acid	Glu	E	129.042593094	129.116158896	$C_5H_7N_1O_3$	6.61
Glutamine	Gln	Q	128.058577516	128.171556905	$C_5H_8N_2O_2$	3.94
Glycine	Gly	G	57.021463726	57.052233860	$C_2H_3N_1O_1$	6.95
Histidine	His	H	137.058911874	137.142140206	$C_6H_7N_3O_1$	2.27
Isoleucine	Ile	I	113.084063982	113.160590603	$C_6H_{11}N_1O_1$	5.92
Leucine	Leu	L	113.084063982	113.160590603	$C_6H_{11}N_1O_1$	9.63
Lysine	Lys	K	128.094963024	128.175293325	$C_6H_{12}N_2O_1$	5.92
Methionine	Met	M	131.040484618	131.197889547	$C_5H_9N_1O_1S_1$	2.38
Phenylalanine	Phe	F	147.068413918	147.178050372	$C_9H_9N_1O_1$	4.00
Proline	Pro	P	97.052763854	97.117549470	$C_5H_7N_1O_1$	4.83
Serine	Ser	S	87.032028410	87.078627759	$C_3H_5N_1O_1$	6.85
Threonine	Thr	T	101.047678474	101.105716944	$C_4H_7N_1O_1$	5.45
Tryptophan	Trp	W	186.079312960	186.215027571	$C_{11}H_{10}N_2O_1$	1.15
Tyrosine	Tyr	Y	163.063328538	163.177355085	$C_9H_9N_1O_1$	3.06
Valine	Val	V	99.068413918	99.133501417	$C_5H_9N_1O_1$	6.73

Table 1.1.: Amino acids with three- and one-letter codes (3-code, 1-code), monoisotopic (monoiso) and average (avg) masses, chemical sum formula and frequency (freq) of occurrence in Swiss-Prot protein sequences. The sum formulas are given for the residues without terminal H and OH groups.

	T		C		A		G	
T	TTT	Phe	TCT	Ser	TAT	Tyr	TGT	Cys
	TTC	Phe	TCC	Ser	TAC	Tyr	TGC	Cys
	TTA	Leu	TCA	Ser	TAA	STOP	TGA	STOP
	TTG	Leu	TCG	Ser	TAG	STOP	TGG	Trp
C	CTT	Leu	CCT	Pro	CAT	His	CGT	Arg
	CTC	Leu	CCC	Pro	CAC	His	CGC	Arg
	CTA	Leu	CCA	Pro	CAA	Gln	CGA	Arg
	CTG	Leu	CCG	Pro	CAG	Gln	CGG	Arg
A	ATT	Ile	ACT	Thr	AAT	Asn	AGT	Ser
	ATC	Ile	ACC	Thr	AAC	Asn	AGC	Ser
	ATA	Ile	ACA	Thr	AAA	Lys	AGA	Arg
	ATG	Met	ACG	Thr	AAA	Lys	ACG	Arg
G	GTT	Val	GCT	Ala	GAT	Asp	GGT	Gly
	GTC	Val	GCC	Ala	GAC	Asp	GGC	Gly
	GTA	Val	GCA	Ala	GAA	Glu	GGA	Gly
	GTG	Val	GCG	Ala	GAG	Glu	GGG	Gly

Table 1.2.: Genetic code DNA→amino acid. STOP denotes stop codons.

2. Protein Identification by Mass Spectrometry

2.1. Probe Preparation

Mass spectrometry has become the method of choice for identification [2, 40] and quantification [101] of proteins in a high-throughput manner.

According to Siuzdak [121], a mass spectrometer is "*an analytical device that determines the molecular weight of chemical compounds by separating molecular ions according to their mass-to-charge ratio (m/z)*". Molecular weights are commonly measured in Dalton (Da) in the biological sciences. One Dalton equals one atomic weight unit (amu) of $1.66 \cdot 10^{-24}$ g, approximately the molecular weight of a single proton.

A typical protein identification procedure using mass spectrometry involves the extraction of the protein mixture from the cell or tissue, the purification of the proteins, the proteolytic digestion by a protease, the measurement of the resulting fragment peptides' masses for each protein of interest, and finally the computational identification either de-novo or by database search. Here, we briefly describe the protein purification and digestion, before we examine mass spectrometry instrumentation in Section 2.2, tandem MS and de-novo identification in Section 2.3 and finally peptide mass fingerprinting in Section 2.4.

Protein purification. Purification of protein mixtures is a two-step process. First, proteins have to be separated from other cell components such as DNA and metabolites; this is usually done using precipitation. Then, the individual protein species in the resulting complex protein mixture have to be separated from each other. We will call this step *protein separation* to distinguish it from the first step.

There are currently two major protein separation techniques suitable for subsequent mass spectrometric analysis: 2D-gel electrophoresis (2D-GE) and liquid chromatography (LC). We will only explain 2D-GE. For more detailed information on both techniques see [105].

2D-gel electrophoresis. A gel consists of cross-linked polymers building a matrix of varying mesh size. In 2D-gel electrophoresis, the separation of proteins is done in two steps. In the first step, the proteins are separated by their isoelectric points using isoelectric focusing. They are loaded onto a matrix with an immobilized pH gradient and a current is applied. The proteins move towards the positive or negative end of the gel, according to their charge. Since the charge changes with pH, a protein will stop moving when it reaches a specific pH value within the gradient, neutralizing its charge.

This first step is usually not enough to separate all proteins and a second step follows, separating the proteins by molecular weight. In this second step, sodium dodecyl sulfate (SDS) is applied to negatively charge the proteins and linearize them; the protein charge depends on its size. An anionic detergent is used to denature the proteins, and a current is applied to the second dimension. Due to the pore size of the polyacrylamide (PA) gel, smaller proteins migrate faster than larger ones, separating proteins of different molecular weight. Gel electrophoresis with SDS and PA is commonly called SDS-PAGE. The proteins are then stained by silver or coomassie blue and made visible in the gel. For mass spectrometric analysis, the spots of interest are picked from the gel either by hand or by a picking robot and each protein species is subjected to mass spectrometry. 2D-gel electrophoresis is a source of artificial mass modifications of proteins. In SDS-PAGE, cysteines are commonly modified by chemical attachment of carbamidomethyl during separation; this alters the mass of the cysteine residues and has to be considered in the following mass spectrometric analysis.

Protein digestion. After separation, the proteins of interest are identified (e.g. by comparison to previous experiments) and biochemically dissociated using a protease. Proteases are hydrolases, i.e., enzymes that cleave a peptide bond by the use of one water molecule (see above). They are usually site-specific, meaning that they cleave peptide bonds between particular residues. Proteases are involved in a myriad of biochemical processes like food digestion and blood clotting. In proteomics, the most commonly used protease is trypsin, which cleaves peptide bonds after each occurrence of an arginine (R) or lysine (K), unless followed by a proline (P). The molecular masses of the resulting cleavage fragments are then measured by mass spectrometry. Trypsin is well-suited for use in mass spectrometry settings since the resulting fragments provide two protonation-sites (the N-terminus and the basic C-terminus) that allow efficient ionization by mass spectrometers (cf. [12]).

2.2. Instrumentation

Mass spectrometric measurements are carried out in the gas phase on ionized analytes [2]. Simply put, a mass spectrometer has three important components:

- An *ion source* for ionizing the analyte,
- a *mass analyzer* for separating the ions by mass-over-charge ratio m/z, and
- a detector for registering the abundance of ions at each m/z value.

We briefly describe the major ion sources and mass analyzers. An extensive treatment with detailed information on instrumentation can be found in [58].

Ion sources. Although mass spectrometry was invented in the late 19th century and used in many chemical and physical applications, the ion sources were not suitable for ionization of large biological molecules. This changed in the 1980s, when two

different "soft" ionization methods, namely matrix-assisted laser desorption/ionization (MALDI) [66,76] and electrospray ionization (ESI) [46,129], were developed that allowed the ionization of intact large biomolecules such as proteins.

In MALDI, the analytes are mixed with a matrix solution and placed on a metal plate after drying. The metal plate is then transfered into the vacuum system of the mass spectrometer and a laser pulse is shot onto the matrix, with wavelength specific to the matrix. Parts of the matrix evaporate, releasing the enclosed analyte and ionizing it. Ionization is mostly singly-charged protonation. Since only a fraction of the analyte and matrix is used with each laser shot, the same probe can be measured multiple times. MALDI plates can also be stored for longer periods, allowing analysis of the same or multiple probes with interruptions. MALDI interfaces well with 2D-GE; it also interfaces with LC if the separated solution is spotted directly onto the MALDI plate.

In ESI, the analyte is dissolved and the solution is pressed through a small, highly charged needle, whereby the analyte is ionized. The resulting small aerosol droplets are sprayed into the vacuum system of the mass spectrometer, where the solution evaporates and the ionized analyte remains. ESI usually produces multiply protonated ions. Unlike MALDI, ESI relies on a constant supply of dissolved analytes. This makes it particularly suited for interfacing with liquid chromatography.

Mass analyzers. Mass analyzers separate the ions according to their mass-to-charge ratio m/z; they are based on the dynamics of ions in an electro-magnetic field. MALDI is usually coupled to time-of-flight (TOF) analyzers, whereas ESI is coupled to quadrupoles or ion traps.

A TOF analyzer accelerates the ionized analyte by applying an electric field for a certain distance. Since the electric field strength, and thus the force applied, are the same for all ions, the velocity of a particular ion after acceleration depends solely on its mass and charge. When coupled to MALDI, all ions can be assumed to be singly charged and thus the velocity directly refers to an ion's mass. After applying the electric field, the ions drift towards the detector in a tube of given length. Hence, the velocity of an ion can be measured by the time it needs to drift from the acceleration area to the detector. Its mass is then calculated. A scheme of a linear MALDI-TOF instrument adapted from [58] is given in Figure 2.1.

A quadrupole consists of four metal rods arranged around the drift tube. A high frequency alternating voltage is applied on the four rods, producing an oscillating electro-magnetic field inside the tube. The ions are now forced into corkscrew-like trajectories within the tube. Depending on the strength and frequency of the applied voltage, only ions within a very narrow mass-to-charge range have a stable trajectory ending in the detector; all other ions are deflected.

Ion traps have a similar principle as quadrupoles, but the ions are trapped within a metal cage. They are forced into specific trajectories within the ion trap by applying an alternating voltage to produce an oscillating electric field inside the trap. Ions of a particular mass-to-charge ratio are then released by applying a different voltage.

Note that all mass analyzers only allow an indirect measurement of mass-to-charge

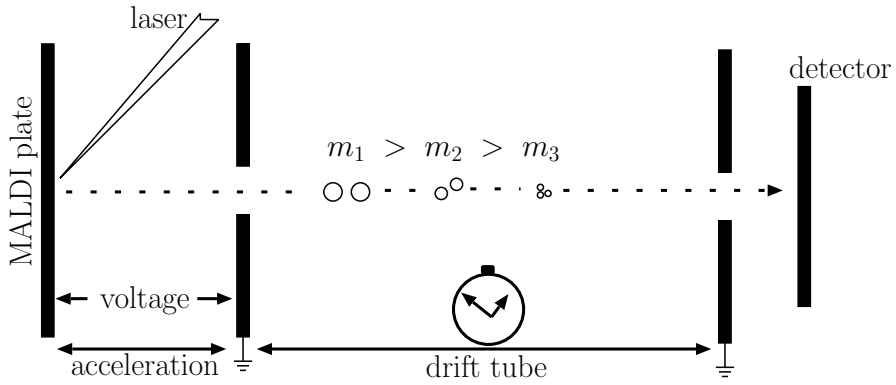

Figure 2.1.: Scheme of a linear MALDI-TOF mass spectrometer (adapted from [58]).

ratios. In order to get m/z values, it is therefore necessary to transform the measured data. In particular, mass spectrometers have to be carefully calibrated to ensure correct transformation.

Ion detectors. The detector is usually built as an electron multiplier, where the ions cause an electric current when hitting the detector. The ion current is recorded in equidistant time steps; it is called the intensity of that particular m/z value as its strength relates to the abundance of ions.

Raw spectra and spectra processing. The measured physical quantity together with the corresponding ion currents measured by the ion detector form the raw spectrum. In a first step, it has to be transformed into m/z values taking into account the calibration parameters. In the following, signal processing algorithms are used for filtering noise, normalizing the signal level ("baseline correction"), and de-isotoping the spectrum by computing mono-isotopic peak masses from observed isotopic patterns. As a final step, peak detection or peak picking algorithms are applied for finding the ion signals in the processed raw spectrum. The detected m/z values together with their measured intensities and possibly other parameters are recorded in a so-called peak list. The peak list is the input data for all further computation and the following identification algorithms.

Algorithms for spectrum processing and peak picking are described, e.g., in [37,81,94].

Mass accuracy. Due to some imperfections in instrumentation, ions of the same m/z value never hit the detector at exactly the same time. For example, before ions are accelerated by an electric field in MALDI-TOF-MS, they already have an initial velocity due to the energy transfered by the laser pulse. Thus, although ions of the same m/z value are accelerated in the exact same way, they nevertheless have slightly different velocities at the end of the acceleration. They hit the detector at slightly different times and are measured as a peak of a certain width. This has two implications: First, ions

of similar m/z values may become indistinguishable and second, the actual m/z value cannot be determined precisely but only within some error range.

Mass accuracy is commonly given in parts-per-million (ppm). With an accuracy of 100 ppm, a molecular mass of 1 000 Da is measured with an error of 0.1 Da. Current MALDI-TOF mass spectrometers have an accuracy of 50 ppm in standard experiments and 10 ppm if carefully calibrated using internal standards [12,33]. For ESI-Quadrupole instruments together with orthogonal acceleration TOF (QqTOF), 5 ppm are reported and Fourier transform (FT) mass spectrometers are routinely used at 1 ppm mass accuracy (both numbers reported in [12]). Note that these mass accuracies can only be achieved with carefully calibration; otherwise, systematic mass errors result.

2.3. Protein Identification

Three major approaches exist for identifying a single protein using mass spectrometry: Peptide mass fingerprinting (PMF), tandem MS (also called peptide fragment fingerprinting, PFF), and de-novo sequencing. While the first two rely on a sequence database for identification and perform a comparison of the measured peak list to peak lists predicted from the sequences, the latter is database independent, although it is usually combined with a database search. Protein identification using very small databases of manually sequenced proteins has been done since the 1980s [71]. With the advent of large genome sequence databases from genome sequencing projects starting in the 1990s, the amount of protein sequences grew dramatically. Together with improved mass spectrometers, this allows protein identification in a high-throughput manner.

For a general overview of identification approaches see the review articles [1,2,7,11,40, 90,102]. The books by Palzill [105] and Snyder [122] provide more detailed information on experimental techniques. A detailed overview of identification algorithms up to 1998 can be found in [134], whereas [102,120] give briefer reviews of more recent developments up to 2005/6. A comparison of the performance of PMF and PFF identification was conducted in [83], revealing a comparable identification rate (97% for PMF and 100% for PFF of 162 gel-separated proteins).

In this thesis, we are exclusively concerned with PMF. Nevertheless, algorithms for PMF and PFF have some similarities and we will thus take a brief look at PFF and de-novo identification before describing identification by PMF in some more detail in Section 2.4.

Peptide fragment fingerprinting by tandem MS. In tandem MS, the m/z values of the peptides from an enzymatic digest of a protein are measured using MS. Each peptide m/z of particular interest is afterwards transfered into a collision chamber, where it collides with a noble gas, and peptide bonds are broken by collision induced dissociation (CID). The m/z values of the resulting new ions are then again measured by MS; they correspond to the prefixes and suffixes of the primary structure of the peptide. The newly measured spectrum is also called the MS/MS spectrum of the peptide. A common instrumentation for tandem MS is LC/ESI coupled to several quadrupole mass analyzers

for selecting peptides of certain m/z, colliding them with a noble gas and measuring the resulting MS/MS spectrum.

For identification, the measured MS/MS peak list is compared to computed peak lists of every peptide in the sequence database and a similarity score is computed for each comparison. The highest scoring peptide is returned as identification. If several peptides from one protein have been identified by MS/MS, the corresponding database sequence is reported as the protein identification. Usually, a significance is computed for each identification. One of the earliest computer programs to identify a peptide by MS/MS is SEQPEP [71]. A standard software is SEQUEST [43,135,136], which computes a cross-correlation of the measured and the predicted MS/MS spectrum, thus implicitly taking into account spectrum quality and peak intensities. Several enhancements have since been proposed, particularly indexing of peptides for faster search [41,87]. The software SCOPE [8] uses a two-step stochastic process to describe fragmentation of peptides, taking into account missing and additional peaks, and instrument mass accuracy. The scoring is done using a dynamic programming algorithm and takes into account the different ionization probabilities (provided manually). A p-value is provided either from an assumed Gaussian score distribution or by a stochastic inequality. ProbID [138] uses a Bayesian scoring scheme to take into account different ion types; it does not provide a score significance. In [125], a scoring model based on a hidden Markov model (HMM) was proposed that takes into account different ion types and ranked intensities. Significance computation is done by estimating the score distribution from 500 sampled peptide sequences of given mass. Parameters of the HMM were estimated on a real proteomics dataset. Other methods include VEMS [93] and OLAV [35,36]. The algorithm proposed in [61] explicitly uses peak intensities and provides a p-value computation.

A comparison of tandem MS identification methods on large-scale proteomics datasets was published in [42], whereas in [28], the performance on well-defined datasets was investigated. In [115], the hypergeometric distribution was proposed for computing statistical significance of identifications.

De-novo sequencing. A database-independent method based on tandem mass spectrometry is de-novo sequencing of the peptide using its MS/MS spectrum. In principle, since the m/z values of each prefix and suffix of the peptide's primary structure are measured, the sequence can be discovered by finding peaks of mass differences of exactly one amino acid. A similar approach was already used in 1984 in [116], where peptide sequences were found by exhaustive search. A manual approach was proposed in [68], involving partial Edman degradation and the sequential use of several proteases. Since then, the problem of peptide sequencing from MS/MS spectra was translated into a so-called spectral graph. Based on this idea, several algorithms have been proposed [29,124]. Other approaches are based on dynamic programming ([9] and PEAKS [89]), and probability networks (PepNovo [49]). The review [88] provides an overview of de-novo algorithms up to 2004.

Usually, de-novo peptide sequencing algorithms are able to sequence only a few amino acids (8–12). In addition to complete sequencing, the algorithms are also used for se-

quencing a small portion of the peptide, the so-called sequence tag, and then performing a sequence database search with these tags. This approach was first proposed in [92] and has since been implemented and enhanced.

Mass spectrometry for DNA/RNA. Mass spectrometry was also successfully used in the context of DNA/RNA for identification and characterization [59, 98, 99, 108], identification of single-nucleotide polymorphisms [21, 114], de-novo sequencing of DNA molecules [20, 22], and differentiation of several strains of bacteria [6].

Proteomics systems. There are various attempts to provide standardized ways of storing, analyzing and exchanging proteomics data sets including raw MS data and identifications. Such approaches include the systems PROTEIOS [51] and ProDB [131]. Recent platforms for proteomics including data warehouses, laboratory work-flow systems, and data standards can be found in [84]. A recent overview of standards and platforms for bioinformatics data in general, including proteomics data, is also given in [109].

2.4. Peptide Mass Fingerprinting

Instead of performing a collision induced dissociation, another common identification technique solely uses the masses of the proteolytic peptides resulting from a protein digest. Whereas the mass of the protein itself is not discriminative enough to identify the protein in a sequence database, its set of peptide masses commonly is. The set of peptide masses is called the peptide mass fingerprint (PMF) or peptide mass maps (PMM) of the protein. It is compared to in-silico computed PMFs of protein sequences from a database, scored, and the highest scoring protein is returned as identification.

Additional and missing peaks. Due to the imperfection of sample preparation, mass spectrometry, and data pre-processing, a sample mass spectrum may differ from an ideal mass spectrum and show *additional* and *missing* peaks.

Additional peaks are sometimes random noise that was wrongly identified by the pre-processing algorithms as an ion peak. These peaks usually have very low intensity. However, additional peaks also occur by detected ions that do not stem from the protein. We could call these *chemical noise*; they can have very high intensities. Predominant chemical noise sources are: (i) Keratin, a peptide found on human and animal skin, and introduced into the sample during preparation [77]. (ii) Matrix ions for ionization using MALDI: Parts of the matrix are destroyed during application of the laser pulse, ionized, and are consequently measured by the MS machine. Usually, these ions have small molecular mass, which makes interpretation of MALDI-TOF-MS spectra below 500 Da difficult [7]. More severely, these matrix ions also tend to cluster to larger ions, although their masses seldomly match the mass of peptide fragments [60]. Studies on the initial velocity of analyte and non-analyte ions using MALDI also revealed that ions with masses exceeding 3 000 Da can be suspected to be matrix clusters [75]

(iii) Since proteases are themselves proteins, they tend to also digest each other, resulting in additional peptide fragment ions. Although these ion masses are known in advance, unexpected peaks may occur if the protease is contaminated and additional cleavage behavior on some sample peptides is observed. This is a known problem of trypsin, that often contains chymotrypsin as contaminant [77]. Although pure trypsin is generally suspected to also perform unexpected cleavage, investigations using high-accuracy Fourier transform MS did not find significant evidence for this [100]. An extensive list of the 100 predominant contaminants and their molecular masses can be found in [39].

Missing peaks occur due to insufficient ionization of the analyte, low abundance of the corresponding peptide or errors in the signal processing/peak detection. In [34], the influence of the matrix solution is investigated for MALDI, and the use of several different matrices has been suggested in [55] for membrane proteins. An overview of tryptic digestion and MALDI is given in [78], where in addition, the exchange of arginine against lysine C-terminals for increased sensitivity has been investigated.

Another problem for PMF identification algorithms are post-translational modifications that alter the mass of a peptide. Moreover, when searching the spectrum against a sequence database, incorrect sequences are a common problem. These might occur due to false prediction of the reading frame when translating DNA sequence data to protein sequences or due to incorrect gene annotation.

The prediction of peptide mass fingerprints using the atomic composition of peptides is investigated in [53] and used in [56]. Prediction of peak intensities using various statistical learning methods is investigated in [54].

An extensive treatment of the strength and weaknesses of mass spectrometry protocols can be found in [12], the reliability of MALDI-TOF PMF identification is investigated in [18]. In [63], a more historic treatment of concepts and methods in PMF can be found.

Identification algorithms. In 1993, five groups published PMF identification algorithms. Three of them [62, 91, 137] use a peak counting score that ranks the sequences by the number of matching predicted peaks within a certain mass window of a measured peak. For one algorithm [69], results were published but the algorithm itself is not explained. Finally, a more advanced scoring scheme called MOWSE (molecular weight search) was proposed in [103]. Based upon this score, two different identification algorithms have been developed, MS-Fit as part of ProteinProspector [33], and MASCOT [106].

The MOWSE scoring scheme takes into account the non-uniform peptide mass distribution. The frequency of occurrence of a particular fragment mass in a protein of given parent mass is estimated from the database as follows: The proteins contained in the database are sorted by their parent mass into bins of 10 kDa mass range. Then, the fragment masses are divided into bins of 100 Da mass range, and the number of occurrences of a fragment mass in a protein is counted for each fragment mass bin and each parent mass bin. For example, one entry of the resulting table is the number of fragments of mass $1\,000 \ldots 1\,100$ Da in proteins of mass $35 \ldots 45$ kDa. Dividing the

number of fragments by the number of proteins in each table entry gives the fragment mass frequency, which is then normalized to the largest value in the table. If a database peptide matches a measured m/z value, the corresponding fragment mass frequency is looked up in the computed table. The frequencies of each matched fragment mass are then multiplied and divided by 50 000 to normalize the score for an average protein of parent mass 50 kDa.

Since the frequencies are estimated from a sequence database, the computed scores depend on the composition and size of the database. In particular, if the sequence database grows, scores computed for a previous version of the database cannot directly be compared with scores of a current version.

The software MASCOT also uses a significance computation for each score and provides a p-value computation, from which a significance score is derived by taking the negative logarithm of a p-value. However, no further details are known about the extensions of the MOWSE scoring.

Another more recent software is ProFound [139]. It is based on Bayesian statistics and takes into account background information such as previous experiments on the same protein, the protein's mass, the available taxonomy, the cleavage enzyme, the mass accuracy and the protein sequence. It assumes a Gaussian mass deviation, provides a many-to-one peak matching and incorporates additional and missing peaks with a simple model. Moreover, ProFound takes into account overlapping and adjacent peptides in its scoring. The probability that the measured spectrum originates from the database protein is returned as the score.

MS-Fit, MASCOT and ProFound were compared in [28], where ProFound was found to be slightly superior to MASCOT and both programs were found to be superior to MS-Fit. None of these three methods takes into account the measured intensities of peaks in the mass spectrum.

In [104], a scoring scheme was proposed that explicitly takes into account the chemical properties of trypsin and various chemical modifications of amino acids. This scoring scheme does also make use of the measured intensities and provides the concept of "pseudo-proteins" to deal with contaminants.

Other PMF identification methods include PepFrag [48] and [56].

Complementary to the identification algorithms, several methods have been proposed to deal with unmatched masses (FindPept [52]), to provide batch filtering and sequential digestion routines [82], and to allow protein identification in complex mixtures [86].

Since many PMF identification algorithms do not provide any significance of scores, several methods for significance computation have been proposed [44, 47, 50]. All of these methods either rely on sampling or on empirical estimation of parameters using an in-silico digest of the sequence database.

2.5. Organization of the Thesis

The aim of the thesis is two-fold: First, we develop a general stochastic model of random peptide mass fingerprints, based on a random model for amino acid sequences and their

molecular masses. In contrast to many existing approaches, such a model will then allow us to compute statistics and estimate significance values under a well-defined random model and independent of a sequence database. A major focus is also the efficient computation of such statistics and significance values within the model. Second, we translate the protein identification problem using peptide mass fingerprints as a global alignment problem. Based on general peak-wise scoring schemes, such alignments provide a general and flexible framework that allows consistent integration of various mass spectrometric features (besides molecular masses) and scores. We will then use the statistics computed from the stochastic model of peptide mass fingerprints to estimate statistical significance of identifications within the alignment framework. Unlike many other approaches, such significances can be interpreted within a well-defined null-model.

The thesis is organized as follows: In the first part (Chapters 3–9), we examine statistics of PMF fragments. After a brief introduction and previous work in Chapter 3, we introduce a mathematical model for random proteins and protein fragments in Chapters 4–5. Our model is based on an extension of weighted alphabets and the combination of weighted strings with standard random models for ordinary strings. The main results in these chapters are weighted hidden Markov models and recurrence equations as computational tools for describing the structure of fragments and deriving fragment statistics.

We examine the distribution of fragment lengths in Chapter 6 and derive the distributions of cleavage sites and number of fragments in Chapter 7. In Chapter 8, we investigate the distribution of fragment masses and related distributions. Finally, we present occurrence probabilities of a fragment mass in a random protein of given length or parent mass in Chapter 9.

Together with the mathematical derivation of the statistics, we present dynamic programming algorithms for computing the statistics and compare our results to corresponding empirical statistics derived from an in-silico tryptic digest of the Swiss-Prot database.

In the second part (Chapters 10–13), we present a general framework for identifying proteins from their peptide mass fingerprints. The framework is based on scoring schemes for peak matching that allow computation of peak list alignments as optimal matchings of two peak lists. After a short introduction in Chapter 10, the framework is formally derived in Chapter 11 and first simple examples of scoring schemes are given. In Chapter 12, we derive the statistical significance of an alignment score by computing its p-value under a well-defined null-model. We demonstrate that the alignment score distribution can be well approximated by a Gaussian distribution and compute the moments of the score distribution. We also give classical inequalities for estimating the p-value for non-Gaussian distributions. We further introduce p-value scores as a general tool for transforming scores of arbitrary scoring schemes into comparable scores. We consider practical aspects of scoring schemes in Chapter 13. We show how mass error distributions and peak intensities can be consistently used in a scoring scheme, and demonstrate the applicability of our framework by comparing identification rates to the standard software MASCOT on real proteomics data.

Finally, we conclude and present some directions of possible future research in Chap-

ter 14.

Parts of Chapters 3–9 have been published in a technical report [73] and are to appear in a refereed conference proceeding [72]. Parts of Chapters 10–13 have been published in a refereed conference proceeding [23] and are to appear in articles in two refereed journals [24, 74].

Part I.

Statistics of Peptide Mass Fingerprinting

3. Introduction

In the first part of the thesis, we develop a general framework based on random weighted strings for computing certain statistics such as fragment length, fragment molecular mass and occurrence probabilities of masses in a random amino acid string. Using the occurrence probabilities, a null model for random mass spectra is induced, based on statistical properties of amino acid molecular masses, the digestion enzyme and the amino acid composition of fragments. This null model will then allow us to estimate score distributions and thus compute p-values of protein identifications in the second part of the thesis.

The first part is organized as follows: In Chapter 4, we introduce a mathematical model for random proteins and application of cleavage enzymes, namely *random weighted strings* and *cleavage schemes*.

In Chapter 5 we investigate the structure of cleavage fragments and introduce *weighted Hidden Markov Models* as a stochastic model of fragments. The statistics of fragmentation is considered in subsequent chapters, where we investigate length distributions of fragments in Chapter 6, joint length-mass distributions of fragments in Chapter 8 and the distribution of cleavage sites as well as the number of fragments in Chapter 7. These statistics are derived using an extension of weighted Hidden Markov Models to *Markov Additive Chains* that also capture the mass of a fragment. In addition, several recurrence and moment equations as well as approximations using standard distributions are presented.

In Chapter 9, the *mass occurrence probability* of occurrence of a certain mass in a random weighted string is introduced and recurrence equations for its computation are derived.

The theoretical results are compared to empirical data derived from an in-silico digest of the Swiss-Prot protein sequence database.

We also give results for the implementation of most statistics that lead to time-efficient algorithms for their computation and to space-efficient storage. We provide estimates of the memory requirements for a standard example.

Related work. Our results can be seen as generalizations of three lines of previous research.

First, our model of probabilistically weighted strings extends the concept of *weighted strings* [30], where the weights of characters are fixed and not probabilistic. Weighted strings have been used in the setting of mass spectrometry for generating peptide candidates [41, 125], for computing possible decompositions of masses into character masses [25, 26] and for finding sub-masses [13, 30]. General combinatorics of weighted strings were investigated in [30].

Second, the waiting times for cleavage points between fragments (i.e., the fragment lengths) are waiting times for specific, possibly overlapping, patterns in strings. For strings without weights, the statistics of such patterns [110, 111, 127] and sets of patterns [27, 112, 113] have been intensively studied in bioinformatics and statistics [133], and our results on random weighted strings naturally contain some of these as special cases.

Third, the model of random weighted strings together with their string mass is a discrete-time variant of a *Markov additive process (MAP)*. MAPs have been intensively studied for continuous-time Markov models and general additive components. Major lines of investigation were existence and limit theorems [31, 32], large deviations and connections to Perron-Frobenius theory [96, 97]. In contrast to MAPs, where the underlying process is an irreducible Markov chain, we will concentrate on i.i.d. sequences as driving processes for the random string.

Notational conventions. We write $\mathcal{L}(X)$ for the distribution of a random variable (r.v.) X; the generic probability measure is denoted by $\mathbb{P}$. Distributions are sometimes represented as probability vectors, e.g., we write $x(m) := \mathbb{P}(X = m)$ for some finite range of integers m and the corresponding probability mass function at m. For a discrete probability distribution $\mathcal{L}(X)$, the probability mass function is sometimes denoted by $\mathcal{L}(X)(m) := \mathbb{P}(X = m)$.

If two random variables X and Y have the same distribution, this is either denoted by $\mathcal{L}(X) = \mathcal{L}(Y)$ or more briefly by $X \stackrel{\mathrm{d}}{=} Y$.

We write $\mathcal{L}(X) \otimes \mathcal{L}(Y)$ for the product measure of $\mathcal{L}(X)$ and $\mathcal{L}(Y)$, i.e., the distribution of the pair (X, Y) if X and Y are independent. Further, $\mathcal{L}(X) \star \mathcal{L}(Y) = \mathcal{L}(X + Y)$ denotes the convolution of the distributions of two independent random variables X and Y. These notations generalize to more than two random variables. The convolution of two vectors $x(i)$, $y(j)$ is defined as $(x \star y)(k) := \sum_i x(i) \cdot y(k - i)$, where the finite value range of k is derived from the ranges of i and j. The convolution of two vectors of sizes n_1 and n_2, respectively, yields a vector of size $n_1 + n_2 - 1$.

For a string s we denote the substring from index i to index j (inclusive) by $s_{i:j}$, and we write $s^{\langle \ell \rangle} := s_{1:\ell}$ for the prefix of length ℓ. To distinguish quantities of the semi-infinite string s from their counterparts of the corresponding finite string $s^{\langle \ell \rangle}$, the latter quantities will also be denoted by an $\langle \ell \rangle$-superscript. The length of a finite string s is denoted by $|s|$. Concatenation of strings s,t is denoted by st.

Further, we write $\bar{\Gamma}$ for the complement of a set $\Gamma \subseteq \Sigma$ within a superset Σ, i.e., $\bar{\Gamma} := \Sigma \setminus \Gamma$. Moreover, $\Gamma \subseteq \Sigma$ includes the case that $\Gamma = \Sigma$, whereas $\Gamma \subset \Sigma$ does not.

Finally, we define the set of natural numbers $\mathbb{N}$ to be the set of positive integers, i.e. it does not include the zero; $\mathbb{N} := \{1, 2, 3, \dots\}$. The set of natural numbers including the zero is denoted $\mathbb{N}_0 := \{0\} \cup \mathbb{N}$.

4. The Random Weighted String Model

Polymers such as proteins or DNA/RNA molecules can be modeled as strings over a finite alphabet such as the alphabet of amino acids of size twenty or the alphabet of nucleotides of size four. In mass spectrometry, we are interested in the molecular mass of such polymers, that is, the sum of masses of its constituent "characters". For modeling this additional quality, we introduce *weighted alphabets*, where each character also carries a *weight* or *mass* represented by a *character mass function*. On these weighted alphabets, we define *weighted strings* together with their *string mass*.

To also capture isotopic mass distributions and mass modifications by post-translational modifications, we extend the concept of weighted alphabets and strings to *probabilistically weighted alphabets and strings*, where we allow a mass distribution for each character.

Both weighted and probabilistically weighted strings are finally combined with standard random string models to derive random weighted strings.

4.1. Weighted Alphabets and Strings

We always assume Σ to be a finite alphabet. The following two definitions restate the concept of weighted strings as used in [30].

Definition 4.1 (Weighted alphabet). Let Σ be a finite alphabet and let $\mu : \Sigma \to \mathbb{N}_0$ be a function assigning each character $\sigma \in \Sigma$ its *mass* or *weight* $\mu(\sigma) := \mu_\sigma$. The pair (Σ, μ) is called a *weighted alphabet* with *character mass function* μ. We write $\mu_{\max} := \max\{\mu(\sigma) : \sigma \in \Sigma\}$ and $\mu_{\min} := \min\{\mu(\sigma) : \sigma \in \Sigma\}$ for the largest and smallest mass in (Σ, μ), respectively, and require $0 < \mu_{\min} \leq \mu_{\max} < \infty$.

Recall that molecular masses are often given in Dalton (Da).

Example 4.2 (Amino acids). Consider Table 1.1 on page 5. Taking column 3 ('1-code') as the finite alphabet and column 5 ('avg mass'), rounded to the next integer, as values for the character mass function gives a weighted alphabet for amino acids with average isotopic masses.

We use nonnegative integer masses for several reasons: Real numbers of arbitrary precision cannot be represented in a computer by standard data types anyway, so it makes sense to restrict masses to numbers in $\mathbb{Q}$ with bounded denominator. By multiplying with an appropriate factor, these can always be represented as integers. For MALDI-TOF mass spectrometry, the mass accuracy of the machine is about 50 ppm [12], so ± 0.1 Da for a mass of 2 000 Da. Multiplying all masses by 10 and rounding to the

nearest integer would then suffice. Even for high-resolution mass spectrometry, such as Fourier-transform MS, the acquired accuracy is about 4 decimals. In addition, in practice, "real" mass values are only known up to a certain accuracy. The conversion factor from integer to "real" masses is called the mass precision.

Definition 4.3 (Mass precision Δ_m). The *mass precision* Δ_m is a factor to convert integer masses into the natural masses given in Dalton. If m^* denotes the mass of a polymer in Dalton, the mass of the corresponding weighted character σ is given by

$$\mu(\sigma) = m = \text{round}(m^*/\Delta_m).$$

The definition carries over to masses derived from character masses.

A mass precision of $\Delta_m = 0.1$ thus gives a scaling factor of 10, so natural masses are represented up to one decimal.

Similarly to alphabets, sequences of elements of a weighted alphabet form so-called weighted strings.

Definition 4.4 (Weighted string, mass process). A *weighted string* over a weighted alphabet (Σ, μ) is an infinite sequence $(s, \mu) = (s_i, \mu(s_i))_{i\in\mathbb{N}}$ over $(\Sigma \times \mathbb{N}_0)^{\mathbb{N}}$.

The sequence $(\mu(s_i))_{i\in\mathbb{N}}$ is called the (deterministic) *mass process* of s. The character mass function is also written with subscripts, i.e., $\mu_{s_i} := \mu(s_i)$.

In subsequent chapters, we will sometimes call the sequence $s = (s_i)_{i\in\mathbb{N}}$ the weighted string without mentioning its mass process.

The case of finite (weighted) strings is obtained by considering the length-ℓ prefix of (s, μ) for some $\ell \in \mathbb{N}$.

The mass process of s is a sequence over $\mathbb{N}_0$; it is not to be confused with the mass of s, which is the sum of its character masses and thus a single number.

Definition 4.5 (String mass). The character mass function is extended to finite strings $s_1 s_2 \cdots s_\ell \in \Sigma^\ell$ for $\ell \in \mathbb{N}$ by setting

$$\mu(s_1 s_2 \ldots s_\ell) \;:=\; \sum_{i=1}^{\ell} \mu(s_i).$$

This extension is then called the *string mass* or simply *mass of string* s.

To model proteins by weighted strings, each amino acid must exactly have one specific molecular mass. This is only true to some approximation: Like all molecules, amino acids have an isotopic distribution. Moreover, post-translational modifications may also alter their mass.

In order to capture isotopic distributions and mass modifications of characters, we want to allow multiple masses per character in an alphabet, where each mass is taken with certain probability. As only the mass is probabilistic and no random model for the character sequences is (yet) assumed, we call such weighted alphabets *probabilistically weighted alphabets*.

Definition 4.6 (Probabilistically weighted alphabet). Let Σ be a finite alphabet, let $(\Xi, \mathbb{P})$ be an appropriately constructed probability space, and let $\mu : \Sigma \times \Xi \to \mathbb{N}_0$ be a *probabilistic character mass function*, assigning to each character $\sigma \in \Sigma$ a random variable $\mu(\sigma, \cdot) = \mu_\sigma(\cdot) : \Xi \to \mathbb{N}_0$, so $\mathbb{P}(\mu_\sigma = m)$ denotes the probability that the mass of character σ takes the value m. The pair (Σ, μ) is then called a *probabilistically weighted alphabet*.

Again we denote by $\mu_{\max}$, $\mu_{\min}$ the largest and smallest possible mass in (Σ, μ), with $\mu_{\max} := \max\{m \in \mathbb{N}_0 : \exists \sigma \in \Sigma \text{ s.t. } \mathbb{P}(\mu_\sigma = m) > 0\}$ and $\mu_{\min} := \min\{m \in \mathbb{N}_0 : \exists \sigma \in \Sigma \text{ s.t. } \mathbb{P}(\mu_\sigma = m) > 0\}$, and require $0 < \mu_{\min} \leq \mu_{\max} < \infty$.

Note that it is sufficient to specify the distribution $\mathcal{L}(\mu_\sigma)$ for each $\sigma \in \Sigma$ and we do not have to explicitly specify the probability space Ξ. Also, if $\mathcal{L}(\mu_\sigma)$ is a Dirac distribution for each $\sigma \in \Sigma$ (in which case μ_σ takes on one value with probability 1), the probabilistically weighted alphabet is the same as a weighted alphabet as we can identify μ_σ with the mass m_σ for which $\mathbb{P}(\mu_\sigma = m_\sigma) = 1$.

Example 4.7 (Isotopes of amino acids). Each amino acid occurs in nature with several masses due to the isotopic masses of the atoms building the amino acid. This fact can be modeled by a probabilistically weighted alphabet, where μ_σ takes on the different isotopic masses of the amino acid σ with probability of occurrence of the corresponding isotopic composition in nature. These probabilities can be computed from the atoms' isotopic distributions, which are known with high accuracy.

Example 4.8 (Post-translational modifications). We can also model post-translational modifications (PTMs) of amino acids with probabilistically weighted alphabets. PTMs can occur at every amino acid or they can be specific to certain amino acids. Thus, some amino acids of certain type may be modified, others may not. Probabilistically weighted alphabets are useful if the frequency of a modification is known or can be estimated for every amino acid. This model can also be combined with the above model of isotopic distributions.

Since we would like to consider strings of arbitrary length in what follows, we develop our models from an infinite string $s \in \Sigma^{\mathbb{N}}$ and then use projections to finite length-ℓ prefixes as needed.

In order to define (probabilistically) weighted strings over a probabilistically weighted alphabet, we first have a look at the sequence of masses associated to a fixed sequence of characters; in contrast to weighted strings, this mass process is now a sequence of random variables, or a stochastic process. We require that the masses of characters at different positions be conditionally independent, given the characters.

We refer the reader to [19] for the basics on stochastic processes.

Definition 4.9 (Mass process for fixed strings). Let (Σ, μ) be a probabilistically weighted alphabet and let $s \in \Sigma^{\mathbb{N}}$ be a fixed infinite string. Let the masses be chosen independently for each character. Then the *mass process* $(\mu(s_i))_{i \in \mathbb{N}}$ is defined as a stochastic process having index set $\mathbb{N}$ and taking values in $\mathbb{N}_0$, where because of independence, the finite dimensional distributions of $(\mu(s_i))_{i \in I}$ for finite $I \subset \mathbb{N}$ are given by

the products

$$\mathcal{L}(\mu_I) := \bigotimes_{i \in I} \mathcal{L}(\mu(s_i)).$$

The double use of μ for both the probabilistic character mass functions $(\mu_\sigma)_{\sigma \in \Sigma}$ and the mass process $(\mu_i)_{i \in \mathbb{N}}$ of a string should not cause confusion, but rather aid intuition. This definition especially extends the mass process of Definition 4.4. It also contains the case of finite strings by restricting I to a subset of $\{1, \ldots, \ell\}$ for fixed ℓ.

The mass associated to a fixed finite string is now also a random variable, as it is the sum of the single random masses. Because of independence, its distribution can be computed as the convolution of the individual distributions.

Lemma 4.10 (String mass distribution). *For finite $I := \{i_1, i_2, \ldots, i_n\} \subset \mathbb{N}$, let $s_I := s_{i_1} s_{i_2} \ldots s_{i_n}$. The distribution of the string* string mass *of s_I is given by*

$$\mathcal{L}\left(\mu(s_I)\right) \;=\; \mathcal{L}\left(\mu(s_{i_1})\right) \star \ldots \star \mathcal{L}\left(\mu(s_{i_n})\right).$$

This is the only reasonable way to consistently extend Definition 4.5: For a (non-probabilistically) weighted alphabet, the distribution of the string mass is again a Dirac distribution, assigning probability 1 to the sum of character masses.

Example 4.11 (String mass). Let $\Sigma = \{a, b\}$ and (Σ, μ) be a probabilistically weighted alphabet with character mass distributions $\mathbb{P}(\mu_a = 1) = \mathbb{P}(\mu_a = 2) = \frac{1}{2}$ and $\mathbb{P}(\mu_b = 1) = \mathbb{P}(\mu_b = 2) = \mathbb{P}(\mu_b = 3) = \frac{1}{3}$. Let $s = ab$. Then the distribution of μ_s is given by

$$\mathbb{P}(\mu_s = m) = \left(\mathcal{L}(\mu_a) \star \mathcal{L}(\mu_b)\right)(m) = \sum_{m' \in \mathbb{N}_0} \mathbb{P}(\mu_a = m', \mu_b = m - m').$$

and we obtain:

m	0	1	2	3	4	5	6
$\mathbb{P}(\mu_s = m)$	0	0	$\frac{1}{6}$	$\frac{2}{6}$	$\frac{2}{6}$	$\frac{1}{6}$	0

4.2. Random Weighted Strings

So far, we did not assume a random model for a string over an alphabet. It is thus not yet possible to capture character frequencies or character dependencies within strings with these models. We show how standard random models for strings can be used to derive random weighted strings.

Random string models are well-known in computational biology. We recapitulate the main definitions and models briefly.

Definition 4.12 (Random string model). A *random string* over an alphabet Σ is a stochastic process S with index set $\mathbb{N}$ taking values in Σ given by its finite dimensional distributions

$$\mathcal{L}(S_I) = \mathcal{L}(S_{i_1}, \ldots, S_{i_n})$$

for all finite $I = \{i_1, \ldots, i_n\} \subset \mathbb{N}$. We require $\mathbb{P}(C = \sigma) > 0$ for each $\sigma \in \Sigma$.

The two major random string models are the independent, identically distributed (i.i.d.) and Markov models.

Example 4.13 (i.i.d. string model). The most simple model for a random string is an *i.i.d. string*, where we assume all characters to be chosen independently from the same distribution. The finite dimensional distributions $\mathcal{L}(S_I)$ then reduce to the product measure $\mathcal{L}(S_I) = \bigotimes_{i \in I} \mathcal{L}(S_i)$. The probability of a string, given the index set I, is the product of its characters' probabilities:

$$\mathbb{P}(S_I = s_I) \;=\; \prod_{i \in I} \mathbb{P}(S_i = s_i).$$

Example 4.14 (Markov string model). A more complex model for a random string is a *homogeneous Markov string of order one*. For its specification, the initial distribution $\mathcal{L}(S_1)$ of the first character and the conditional distribution $\mathcal{L}(S_2|S_1)$ are needed. Homogeneity then means that this conditional character distribution is the same for all subsequent characters.

Combining a standard random string model with a (probabilistically) weighted alphabet, we are now able to give a general random string model for weighted strings over both deterministically and probabilistically weighted alphabets. In all cases, we assume that the mass of a character in a string is independent of the masses of all other characters.

Definition 4.15 (Random weighted string). A *random weighted string* is a stochastic process $(S, \mu) = ((S_1, \mu_1), (S_2, \mu_2), \dots)$ with index set $\mathbb{N}$, values in $\Sigma \times \mathbb{N}_0$ and finite dimensional distributions

$$\mathcal{L}((S, \mu)_I) = \mathcal{L}(S_I) \otimes \mathcal{L}(\mu_I)$$

where S is a random string and μ is a mass process associated to S.

Henceforth, we exclusively discuss the i.i.d. model, in which we assume the characters to be independent and identically distributed. Note, however, that all above definitions also capture arbitrary random string models, the most prominent one being Markov sequences.

Example 4.16 (Random proteins). As before, we model a peptide or a protein as a string over the alphabet Σ of amino acids, and every character σ in the string has a certain mass μ_σ dependent only on the character itself, but independent of all other characters within the string. The character 'L' $\in \Sigma$, say, at some given position within the sequence may therefore have a mass different from that of the same character 'L' later in the sequence. A useful random peptide model would take the frequencies of amino acids from a sequence database such as Swiss-Prot as character probabilities (see Table 1.1). The isotopic character mass distributions can be computed from the isotopic distributions of the atoms by convolution.

If the character probabilities $\mathbb{P}(C = \sigma)$ and the distributions $\mathcal{L}(\mu_C \mid C = \sigma)$ are known for each character σ, we can use the identity $\mathbb{P}(C = \sigma, \mu_C = m) = \mathbb{P}(\mu_C = m \mid C = \sigma) \cdot \mathbb{P}(C = \sigma)$ to obtain the joint character-mass-distributions. From these, the mass distribution of the whole string can be computed.

5. Fragmentation of Random Weighted Strings

Recall that for identifying a biomolecule such as a protein by mass spectrometry, the mass of the biomolecule itself is of little value. In most mass spectrometry settings, the molecule is therefore cleaved into fragments using a biochemical cleavage reaction. In the case of proteins, these so-called proteases usually cleave the amino acid sequence right after the occurrence of a specific amino acid. For some proteases, however, this cleavage reaction is suppressed if another specific amino acid occurs right after the potential cleavage site. In what follows, we will exclusively consider proteins and peptides and their fragmentation by proteases. Note, however, that the model is also valid for some RNAses for cleaving DNA/RNA sequences.

For modeling the action of proteases, we introduce *cleavage schemes* together with corresponding semantics. Applying a protease to a randomly chosen protein is modeled by applying a cleavage scheme on a random weighted string, resulting in *fragmentation* of this string. The string masses of the *fragments* are then a model of the mass fingerprint of the original string.

Our discussion first focuses on infinite random strings $S \in \Sigma^{\mathbb{N}}$ to avoid complications with boundary effects; the necessary adjustments for finite strings are made subsequently.

5.1. Cleavage Schemes and Fragmentation

To formalize the parameters of the described proteases, we introduce the concept of a *cleavage scheme* which is formed by the sets of *cleavage characters* and *prohibition characters*.

It is assumed that cleavage takes place after the cleavage character. Some enzymes, however, cleave before the cleavage character. If this cleavage reaction is suppressed by a prohibition character before the cleavage character, all following statistics remain valid for finite string in the i.i.d. string model, as we can just consider the reversed strings.

Definition 5.1 (Cleavage scheme (Γ, Π); quantity p_Θ). A *cleavage scheme* is a pair (Γ, Π) of a set of *cleavage characters* $\Gamma \subset \Sigma$, and a set of *prohibition characters* $\Pi \subset \Sigma$. To exclude the pathological case that no cleavage takes place, we also require $\Gamma \neq \emptyset$.

If the additional constraint $\Gamma \cap \Pi = \emptyset$ (i.e., $\Gamma \subset \bar{\Pi}$) holds, we speak of a *standard cleavage scheme*.

Cleavage schemes with $\Pi = \emptyset$ are called *simple*; every simple scheme is also a standard scheme.

Strings $P = P_1P_2 \in \Gamma \times \bar{\Pi}$ are called *cleavage patterns* since the cleavage reaction takes place within them. For simple cleavage schemes, we can neglect the second character of the cleavage pattern, since $P_2 \in \bar{\Pi} = \Sigma$ in this case; the cleavage pattern then has length 1.

We set $p_\Theta := \mathbb{P}(S_i \in \Theta)$ for any $\Theta \subseteq \Sigma$. In particular, $p_\Gamma := \mathbb{P}(S_i \in \Gamma)$, $p_\Pi := \mathbb{P}(S_i \in \Pi)$. For further use, we note that for standard schemes, $\mathbb{P}(S_i \in \Gamma \cap \bar{\Pi}) = p_\Gamma$ and that $\mathbb{P}(S_i \in \bar{\Gamma} \cap \bar{\Pi}) = 1 - (p_\Gamma + p_\Pi)$.

The stochastics of simple schemes are considerably more straightforward than stochastics for non-simple schemes.

Example 5.2 (Simple schemes: Lys-C and Pepsin). The protease Lys-C cleaves after lysine (1-letter-code K), the protease Pepsin after phenylalanine (F) and leucine (L). Both reactions are not suppressed by any other amino acid; they refer to simple schemes.

The reason for introducing the special case of standard cleavage schemes is that many existing enzymes follow this form, and computations are simplified when compared to general cleavage schemes.

Example 5.3 (Standard scheme: Trypsin). For the frequently used protease Trypsin, we have a cleavage reaction after K or R, if not followed by P, thus $\Gamma = \{K, R\}$ and $\Pi = \{P\}$; it is a standard cleavage scheme. The possible cleavage patterns are of the form $P_1P_2 \in \{K, R\} \times (\Sigma \backslash \{P\})$.

There are also proteases whose cleavage reaction cannot be modeled as a standard scheme.

Example 5.4 (Non-standard scheme: Glu-C (acidic)). The protease Glu-C (acidic) cleaves after the occurrence of D or E. The cleavage reaction is suppressed by a following D or E. As the two sets are not disjoint, Glu-C is not a standard scheme.

Further examples of proteases that match our definition of cleavage schemes, and also an exception, can be found in Table 5.1.

Example 5.5 (Standard example TryptSwissProt). Throughout this thesis, we use tryptic digestion with $\Gamma = \{K, R\}$ and $\Pi = \{P\}$ as a standard example. Character probabilities are estimated as frequencies from all proteins contained in the Swiss-Prot database, release 48 as of September 2005 [10]. These probabilities are listed in Table 1.1, last column. In particular, we estimate $p_\Gamma = 0.1125$ and $p_\Pi = 0.0483$. We use a mass precision of $\Delta_m = 0.1$, so masses are scaled and rounded to cover the first decimal.

We refer to this setting as TryptSwissProt.

Applying a cleavage scheme on a string results in a fragmentation of this string in consecutive, non-overlapping substrings, the fragments. Start and end-indices of these fragments in the string are given by the occurrences of the cleavage pattern.

Recall that for a given cleavage scheme, cleavage occurs after each occurrence of a cleavage character. Further recall that for non-simple schemes, cleavage is suppressed when a prohibition character follows directly after the potential cleavage site.

Protease	Cleaves after	except before	standard scheme
arg-C	R	P	+
asp-N	before D		+
chymotrypsin	E,(L, M),W,Y	P,after PY	- (PY)
cyanogen bromide	M		+
Glu-C (basic)	E	P or E	- $(\Gamma \cap \Pi \neq \emptyset)$
Glu-C (acidic)	D or E	D or E	- $(\Gamma \cap \Pi \neq \emptyset)$
Lys-C	K		+
pepsin (high activity)	F or L		+
pepsin (low activity)	A, E, F, L, Q, W, Y		+
proteinase-K	A, C, F, G, M, S, W, Y		+
trypsin	K or R	P	+

Table 5.1.: Proteases and their cleavage behavior

Definition 5.6 (Cleavage process; cleavage points). Let S be a (random or fixed) infinite string over Σ. Each element of the sequence $(C_i(S))_{i\in\mathbb{N}_0}$ of cleavage pattern occurrences in S with $C_0(S) := 0$ and

$$C_i \equiv C_i(S) := \min\{k > C_{i-1}(S) : S_k \in \Gamma, S_{k+1} \in \bar{\Pi}\}$$

is called a *cleavage point* of S. The series $(C_i(S))_{i\in\mathbb{N}_0}$ is called the *cleavage process* of S. We define $C_i(S) := +\infty$ if the minimum is taken over the empty set. We write C_i short for $C_i(S)$ if S is given from the context.

Note that for simple schemes, the cleavage points are exactly the occurrences of a cleavage character in the string, since with $\bar{\Pi} = \Sigma$, the condition $S_{k+1} \in \bar{\Pi}$ is always satisfied.

The resulting fragments can be described in terms of the cleavage points: A fragment is a substring that starts right after and ends at a cleavage point.

Definition 5.7 (Fragments; fragmentation). For each $i \geq 1$, the substring $F_i := S_{C_{i-1}+1:C_i}$ is called the *i-th fragment* of S. We denote the length of fragment F_i by $L_i := C_i - C_{i-1}$. The family $(F_i)_{i\geq 1}$ is called the *fragmentation* of S.

Of course, real proteins do not have infinite length, so we need to extend our definitions to the case of finite strings.

Definition 5.8 (Cleavage points for finite strings; fragmentation size). For finite length prefixes $S^{\langle\ell\rangle}$, we define

$$C_i^{\langle\ell\rangle} := \min\{C_i, \ell\},$$

so that potentially all cleavage points lie directly behind the end of the prefix. If $C_{i-1}^{\langle\ell\rangle} = C_i^{\langle\ell\rangle}$, the i-th fragment and the following fragments are empty.

Analogously, we define $F_i^{\langle\ell\rangle}$, $L_i^{\langle\ell\rangle}$, and the fragmentation in terms of $C_i^{\langle\ell\rangle}$. The – now finite – size of the fragmentation is denoted $N^{\langle\ell\rangle} \equiv N^{\langle\ell\rangle}(S)$.

There is an important difference between the fragmentation of the two strings $S_{1:\ell}$ and $S^{\langle\ell\rangle}_{1:\ell}$: Let C_k be the last cleavage point smaller or equal ℓ (so we also have that $C_k = C^{\langle\ell\rangle}_k$). Assume first that $C_k < \ell$; then the suffix $S^{\langle\ell\rangle}_{C_k+1:\ell}$ of the finite string is a fragment, whereas its counterpart $S_{C_k+1:\ell}$ in the infinite string is not. This makes intuitive sense: The remaining part of a protein after the last cleavage site should be treated as a fragment, as it is cleaved from the rest of the molecule and appears in the mass fingerprint. Note that this last fragment is not contained in the fragmentation of S, so we cannot simply take the latter fragmentation, identify all fragments which end before ℓ and take these as fragmentation of $S^{\langle\ell\rangle}$. For $C_k = \ell$, however, the two fragmentations are identical up to string position ℓ.

Figure 5.1 provides a visualization of the fragmentation quantities. Fragment $F^{\langle\ell\rangle}_3$ starts with a cleavage character and $F^{\langle\ell\rangle}_4$ contains a cleavage character inside; both are neutralized by the following prohibition characters. The last fragment $F^{\langle\ell\rangle}_4$ may end with any character.

Example 5.9 (Fragmentation of a string). Let $\Sigma := \{X, C, P\}$, $\Gamma := \{C\}$, $\Pi := \{P\}$ and let $s = PCCPXXCPXCCC$ be a fixed finite string of length $\ell = 12$. The cleavage patterns are CC and CX, and $\mathcal{F}^{\langle\ell\rangle}_S = (PC, CPXXCPXC, C, C)$ is the fragmentation of s of size $N^{\langle\ell\rangle} = 4$, cleavage points $C^{\langle\ell\rangle}_1 = 2$, $C^{\langle\ell\rangle}_2 = 10$, $C^{\langle\ell\rangle}_3 = 11$, $C^{\langle\ell\rangle}_4 = 12$, and fragment lengths $L^{\langle\ell\rangle}_1 = 2$, $L^{\langle\ell\rangle}_2 = 8$, $L^{\langle\ell\rangle}_3 = 1$, $L^{\langle\ell\rangle}_4 = 1$. For $i \geq 5$, we have $C^{\langle\ell\rangle}_i = 12$ and $L^{\langle\ell\rangle}_i = 0$.

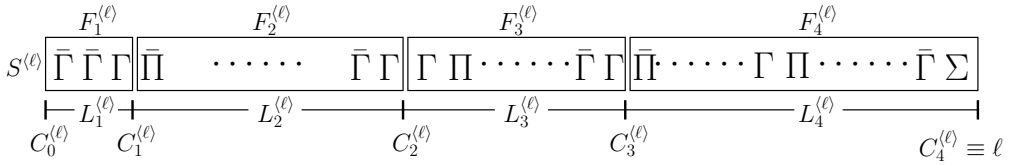

Figure 5.1.: Fragments $F^{\langle\ell\rangle}_i$, cleavage points $C^{\langle\ell\rangle}_i$ and fragment length $L^{\langle\ell\rangle}_i$ of a string $S^{\langle\ell\rangle}$

String fragmentation as regenerative process. Cleavage points in a random weighted string under a cleavage scheme define recurrent events. Fragmentation of random weighted strings can therefore be seen as a renewal process, see e.g. [45, Chapters XIII and XIV] for an introduction.

For infinite strings, the cleavage process $(C_i)_{i\geq 1}$ defines a renewal sequence with delay C_1 and inter-renewal sequence $(L_i)_{i\geq 2}$, since $C_i = C_{i-1} + L_i$ for $i \geq 2$; see Figure 5.1. The cleavage characters not followed by a prohibition character therefore form a regenerative process. Note that the delay corresponds to the length of the first fragment and that all following fragments have the same length distribution $\mathcal{L}(L_2)$ in an infinite string S.

For a simple cleavage scheme ($\Pi = \emptyset$), the renewal sequence can be seen as a non-delayed sequence starting with index 0. In this case, the length distributions of the first and second fragment are the same, and the C_i are stochastic stopping times.

For non-simple cleavage schemes, the next character has to be considered to decide whether the string is cleaved at a specific position. The cleavage points are then no longer adapted to the string process; they are not stopping times.

For finite strings $S^{\langle \ell \rangle}$, we have to deal with a stopped renewal sequence. Here, the length distribution of fragment i depends on the remaining string length $\ell - C_{i-1}$.

5.2. Terminal-Extended Alphabets and Weighted Hidden Markov Models

Terminal-extensions. There is still an important aspect of protein cleavage fragments missing in the model: When a peptide bond is opened by a cleavage enzyme, the new $N-$ and $C-$ termini are again "completed" by attaching an H and OH group, respectively. This alters the average mass of the fragment by approximately 18 Da. Additionally, when measuring such a peptide in a mass spectrometer, the peptide is ionized first. Common ionization methods like MALDI use protonation, which adds another proton of mass about 1 Da to the peptide. Thus, in a MALDI-TOF experiment, all protein fragments will have an average mass of 19 Da above their respective string mass. The main problem for the model is the fact that these additional masses are neither present in the original string nor captured by the weighted alphabet. We cope with this by introducing *terminal-extended weighted alphabets and strings*.

Definition 5.10 (Terminal-extended weighted alphabet). A *terminal-extended (probabilistically) weighted alphabet* is a (probabilistically) weighted alphabet (Σ, μ) together with two *terminal characters* ε_s and ε_e, each of length 0, with weights $\mu(\varepsilon_s)$ and $\mu(\varepsilon_e)$. We explicitly allow one or both of these masses to be zero; both masses are not considered when computing $\mu_{\min}, \mu_{\max}$ of the weighted alphabet.

Let S be a weighted string. Then a *terminal-extended weighted string* is a string $S' = \varepsilon_s S \varepsilon_e$, of length $|S'| := |S|$ and mass $\mu(S') := \mu(\varepsilon_s) + \mu(S) + \mu(\varepsilon_e)$.

The terminal characters can be included in a random model on S to get a random model on S', but we will expect them to have probability one in their respective position. The terminal-extended weighted string S' no longer follows an i.i.d. model, whereas S is unaffected.

Note that this definition also includes mass distributions for the terminal characters and we can use isotopic distributions for the terminal chemical groups if necessary.

For modeling terminal chemical groups common to all fragments, the first terminal character ε_s should be used exclusively to ease projection of a random weighted string S to a finite prefix-string $S^{\langle \ell \rangle}$. We will study this issue further in Section 8.1.

Terminal-extended alphabets and strings are best seen as a purely formal description and are consequently circumvented in practical computations. As the terminal characters are fixed and are the same for each fragment, and they do not contribute to the fragment

length, we can simply adjust the fragments by the terminal characters after performing all necessary computations.

Weighted HMMs. Deterministic and stochastic automata are a common tool to describe strings that obey certain constraints.

In that context, fragments can also be considered as weighted strings obeying certain constraints imposed by the cleavage scheme. We introduce the framework of *weighted Hidden Markov Models* that describe protein fragment sequences in their state sequences and the associated mass processes in their output sequences.

Definition 5.11 (Weighted Hidden Markov Model (wHMM)). A *weighted Hidden Markov Model (wHMM)* is a 6-tuple $(E, P, p^0, T, (\Sigma, \mu), Q)$ consisting of

- a finite set of states E,
- a (sub)-stochastic state transition matrix $P = (P_{ij})_{i,j \in E}$,
- an initial state distribution p^0,
- a set of final states $T \subset E$,
- an (terminal-extended) input weighted alphabet (Σ, μ), and
- a matrix $Q = (q_i(m))_{\substack{m \in \mathbb{N}_0 \\ i \in E}}$ of output distributions $Q_i = (q_i(m))_{m \in \mathbb{N}_0}$ of character masses for each state i,

where the 3-tuple (E, P, p^0) is a homogeneous (defective) Markov Chain.

Weighted HMMs can be interpreted as generating a random fragment string together with its mass process. The semantics are as follows: The state set E is derived by first introducing one state per character of the (terminal-extended) input alphabet Σ. After connecting the states by transitions defined via the transition probability matrix P, the state space may be reduced by joining states with equivalent transitions; these new states now correspond to sets of characters. Each state in E is associated with such a subset of the input alphabet Σ; it represents this set. To avoid cumbersome notation, we enumerate E and identify each state with its ordinal number. This allows us to speak of state i, for example, without writing down its defining set of characters. We do not require character sets to be pairwise disjoint: Two states may share a set of characters. Note that for the assumed i.i.d. random string model, both terminal characters get a singleton state $\{\varepsilon_s\}$ and $\{\varepsilon_e\}$, the latter necessarily being the only final state. We can then call $\{\varepsilon_s\}$ the *start state*, as the fragment will start here with probability 1. We will see examples of wHMM construction for given cleavage schemes in Section 5.3.

For the moment, let W_k denote the state of the wHMM at time k. The start state W_0 is picked according to the start distribution p^0, with $p^0_i = \mathbb{P}(W_0 = i)$. A transition to a new state, being in state i, is made according to the probability distribution in row i of P. The entries of the transition matrix are given by $P_{ij} = \mathbb{P}(W_{k+1} = j \mid W_k = i)$

for all $k \geq 0$; these are homogeneous transition probabilities. For states $i \in T$, these distributions are defective (i.e., they sum to zero and not to one); the wHMM halts in these states. The transition matrix is computed from the underlying cleavage scheme, see Section 5.3 for details.

The sequence of states taken by a run of the wHMM thus forms a *Markov chain* W with transition matrix P, initial distribution p^0 and state space E.

When state i is entered after a transition, a character mass μ is output according to the character mass distribution Q_i of state i. The matrix Q contains the mass emission probabilities for each state; it has $|E|$ columns. The number of rows is arbitrary to some extend: We need a minimum of $\mu_{\max} - \mu_{\min} + 1$ rows to capture the whole mass range of the weighted alphabet (Σ, μ). For computational purposes that will become evident in Section 8.5, we define Q to have $\mu_{\max} + 1$ rows, ignoring terminal characters, and start the indexing of masses at minimal mass $m = 0$.

We assume that cleavage occurs before entering T, but we do allow masses to be emitted in these states to model certain chemical groups at the end of a fragment. As mentioned before, modeling additional masses in a final state may cause problems when computing statistics on prefixes of such fragments; these additional masses should be modeled in a start state to ensure they are also accounted for in a stopped wHMM. To derive the character mass distributions Q_i of states from the weighted alphabet and the cleavage scheme, we extend the alphabet's character mass function μ to states and character sets.

Definition 5.12 (Mass function of character sets; quantity Σ_i). Let $(E, P, p^0, T, (\Sigma, \mu), Q)$ be a wHMM. The distribution $\mathcal{L}(\mu(\sigma))$ of the character mass $\mu(\sigma)$ can be extended to sets of characters $\Theta \subseteq \Sigma$ as follows:

$$\mathbb{P}(\mu(\Theta) = m) \;:=\; \mathbb{P}(\mu(C) = m \mid C \in \Theta) \;=\; \sum_{c \in \Theta} \frac{\mathbb{P}(C = c)}{\mathbb{P}(C \in \Theta)} \cdot \mathbb{P}(\mu(c) = m),$$

where C is a random character. If we denote by Σ_i the set of characters defining state i, the character mass function is also canonically extended to states of the wHMM and the output distributions Q_i for each $i \in E$ are the conditional probabilities

$$q_i(m) \;:=\; \mathbb{P}(\mu(\Sigma_i) = m).$$

The distribution Q_i of a state mass is a mixture of the involved character mass distributions with mixture coefficients $\mathbb{P}(C = c \mid C \in \Sigma_i)$. This is a natural extension of μ: If state i represents only one character σ, then we get the original definition of μ: $\mathbb{P}(\mu(\Sigma_i) = m) = \mathbb{P}(\mu(C) = m \mid C \in \{\sigma\}) = \mathbb{P}(\mu(\sigma) = m)$.

Let again W_k be the state of the wHMM at time k. We can imagine a two-step process: First, a character x is chosen from the character set Σ_{W_k}. Then, the character mass $\mu(x)$ is emitted according to the character mass function of character x.

Let us denote by S_k the character chosen at step k. Then the sequence $S = (S_1, S_2, \dots)$ is a random string. The associated sequence $(\mu(S_1), \mu(S_2), \dots)$ of emitted masses is the mass process of S. If L is the random number of steps just before the wHMM arrives in

a final state, the string $S_1 \dots S_L$ together with its mass process represents a fragment. The fragment is thus defined by the wHMM which itself is constructed from the weighted alphabet and the cleavage scheme. Adding up the individual character masses given by the mass process yields the fragment mass; this will be further investigated in Section 8.1.

Weighted Hidden Markov Models can be interpreted as special cases or extensions of various existing concepts: In a stochastic context, they are related to general Markov Additive Processes, with finite discrete state and output spaces. They can also be interpreted as ordinary Hidden Markov Models, with (E, P, p^0) as underlying Markov Chain, $\mathbb{N}_0$ the output alphabet and $q_i(\cdot)$ the emission probabilities. As we assume the character weight distributions $\mathcal{L}(\mu(\sigma))$ to be of finite support for each $\sigma \in \Sigma$, the output alphabet is necessarily finite and can be restricted to the set $[\mu_{\min}, \mu_{\max}] \subset \mathbb{N}$ (ignoring terminal characters). In this context, the sequence of states visited by the wHMM is the sequence of chosen characters in each visited state, i.e. the string. The observed sequence is the mass process of that particular string. We can now apply standard algorithms for HMMs such as the Viterbi algorithm for computing the most probable output sequence and the most probable hidden state sequence, given the observed output.

In [73], wHMMs were defined slightly different: The output was defined as a weighted character $(S_i, \mu(S_i))$ and not just its mass.

In a computer science context, wHMMs can be interpreted as probabilistic finite automata for generating certain (weighted) strings; they can also be seen as transducers.

As we will show in the next sections, it is straightforward to construct fragmentation wHMMs for the i.i.d. string model and general cleavage schemes. However, the framework of wHMMs is much more general: We can construct wHMM models for more complicated cleavage rules, or for Markovian string models, still using the same computational framework.

5.3. Structure of Fragments

Before investigating the statistics of fragmentation, let us first examine the combinatorial structure of fragments. This structure can be described in two equivalent ways: Either by using wHMMs, giving us the opportunity to investigate the construction of a wHMM for a given cleavage scheme, or by using feasible sets of strings, ultimately leading to efficient dynamic programming algorithms. We will ignore the terminal characters in the description of feasible sets; they are easily incorporated in the statistics afterwards and just obscure the exposition.

We proceed in a bottom-up manner, first investigating simple schemes, moving further to standard schemes, and finally to general cleavage schemes. We denote the corresponding fragments as simple, standard or general fragments. In subsequent chapters, we give results top-down, stating the general results first and then making specializations.

Preliminaries. To ease later expositions, let us first spend some time on a few preliminary considerations.

For non-simple cleavage schemes, the first fragment in a fragmentation has a different structure than the following ones. Whereas the first fragment may start with any character, any following fragment must necessarily start with a non-prohibition character. In an infinite string, all following fragments are of the same structure. If we only consider combinatorial or distributional properties of fragments, we may therefore only distinguish these two different types of fragments and introduce the following simplified notation: As before, we denote the first fragment by F_1. Since all following fragments have the same distributional properties, in particular $F_i \overset{\mathrm{d}}{=} F_j$ and $\mu(F_i) \overset{\mathrm{d}}{=} \mu(F_j)$ for all $i, j \geq 2$, we introduce a new random weighted string F_+, with these properties, i.e. $F_+ \overset{\mathrm{d}}{=} F_2$ and $\mu(F_+) \overset{\mathrm{d}}{=} \mu(F_2)$ in particular. Further, we introduce a new random variable $L_+ \overset{\mathrm{d}}{=} L_2$ to denote the length of "the" fragment F_+. We may imagine that the $+$ symbol is a wild-card for one of the numbers $2, 3, \ldots, N^{\langle \ell \rangle} - 1$. As a last new notation, let $\circ$ denote any of the symbols $\{1, +\}$.

The notations carry over to their counterparts $F_1^{\langle \ell \rangle}$ and $F_+^{\langle \ell \rangle}$ in the case of a finite string. In this case, the following fragments are no longer i.i.d., since their distribution now depends on the remaining string length. However, their distribution depends solely on this remaining length: We have that $F_i(S^{\langle \ell \rangle}) \overset{\mathrm{d}}{=} F_j(S^{\langle \ell' \rangle})$ for $i, j \geq 2$, given that the number of characters after the $(i-1)$-th and the $(j-1)$-th fragment in $S^{\langle \ell \rangle}$ and $S^{\langle \ell' \rangle}$, respectively, is the same. The same observation holds for the masses and lengths of these fragments. We can thus restrict our investigations on distinguishing between the first and a following fragment, always for a given remaining string length.

We make the following definitions to formally introduce the sets of feasible strings:

Definition 5.13 (Feasible sets of fragment strings). For any cleavage scheme (Γ, Π), the *sets of feasible first and following fragment strings of length l* are defined as

$$\mathcal{F}_1(l) := \left\{ f \in \Sigma^l \,|\, \exists s \in \Sigma^{\mathbb{N}} : F_1(s) = f \right\},$$
$$\mathcal{F}_+(l) := \left\{ f \in \Sigma^l \,|\, \exists s \in \Sigma^{\mathbb{N}} : F_i(s) = f \text{ for some } i \geq 2 \right\},$$

respectively.

For the case of finite remaining string length ℓ, let

$$\mathcal{F}_1^{\langle \ell \rangle}(l) := \left\{ f \in \Sigma^l \,|\, \exists s \in \Sigma^{\mathbb{N}} : F_1^{\langle \ell \rangle}(s) = f \right\},$$
$$\mathcal{F}_+^{\langle \ell \rangle}(l) := \left\{ f \in \Sigma^l \,|\, \exists s \in \Sigma^{\mathbb{N}} : F_i^{\langle k+\ell \rangle}(s) = f,\, C_{i-1}^{\langle k+\ell \rangle} = k, \text{ for some } i \geq 2 \right\},$$

for any $k \in \mathbb{N}$.

Simple cleavage schemes. Simple fragments are substrings bounded by occurrences of cleavage characters and the string boundary. One single fragment $F_\circ$ of length l in an infinite string is a sequence of $l-1$ non-cleavage characters followed by exactly one cleavage character and we have $F_1 \overset{\mathrm{d}}{=} F_+$. In a finite string, the last fragment is also a sequence of $l-1$ non-cleavage characters but may end with an arbitrary character.

Fragments of simple schemes in infinite strings can be described using a wHMM as shown in Figure 5.2. The formal definition of this wHMM is as follows: We take a terminal-extended weighted alphabet of amino acids with any character weight function as given e.g. in Table 1.1, and two terminal characters $\varepsilon_s, \varepsilon_e$. We model all mass modifications by terminal groups at the $N-$ and $C-$ termini and the protonation of the fragment in the terminal character ε_s, whereas ε_e has mass 0. This way, we avoid difficulties when stopping the wHMM after a given finite number of steps to generate fragments in a finite string.

We then generate one state for each character. The state $\{\varepsilon_s\}$ is the only start state and $\{\varepsilon_e\}$ the only final state. For each cleavage character, the only transition is into the final state. For each non-cleavage character, we may either transit to another non-cleavage character or transit to a cleavage character. We thus immediately identify Γ and $\bar{\Gamma}$ as the relevant subsets of Σ. The reduced state set E then consists of four states $0, \ldots, 3$, where $T = \{3\} = \{End\}$ is the only final state and the state 0 (*Start*) corresponds to the mass modifications by terminal masses and protonation. Neither contributes to the length of the fragment. The initial distribution, $p^0 = (1, 0, 0, 0)$ is a Dirac distribution forcing the model to start in the start state 0. Once the model is in state 2, the fragment ends with a transition to the final state 3. The transition probabilities are easily derived: Since the random weighted string follows an i.i.d. model, each transition probability is the probability of the character set it enters. Note that we did not require a wHMM to have a distinguished start state; such a state is only necessary if we want to model terminal characters. If no terminal characters are required, we can either nevertheless introduce a start state that does not emit any mass, or we can renounce to use a special start state and adapt the initial state distribution to ensure that the first state corresponds to a character set feasible for the fragment.

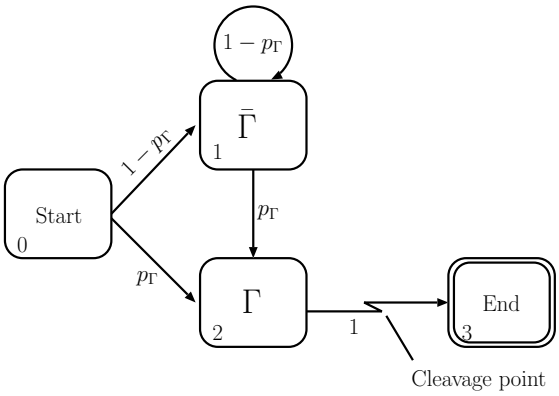

Figure 5.2.: wHMM for simple schemes. See text for details.

Fragments of simple schemes can also easily be described by feasible sets of strings: Let $\mathcal{F}_\circ(l) \subseteq \Sigma^l$ denote the set of fragment strings of length l in an infinite string. Let $\mathcal{F}_\circ^{\langle \ell \rangle}(l)$ denote its counterpart for remaining finite string length ℓ; it is the set of all fragments whose lengths are bounded by ℓ, no matter the character at this position.

Then, $\mathcal{F}_\circ(l) = \bar{\Gamma}^{l-1} \times \Gamma$, as any fragment of length l consists of $(l-1)$ non-cleavage characters followed by a cleavage character. For finite remaining string length ℓ, a fragment of length $l < \ell$ has the same structure as for infinite remaining string length; there must be at least one fragment following it. Thus, $\mathcal{F}_\circ^{\langle \ell \rangle}(l) = \mathcal{F}_\circ(l)$ for $l < \ell$. If a fragment hits the end of the string, its last character is arbitrary. To reach the end of the string, the fragment must be as long as the remaining string, so $l = \ell$. It must not contain a cleavage character inside. Thus, $\mathcal{F}_\circ^{\langle \ell \rangle}(\ell) = \bar{\Gamma}^{\ell-1} \times \Sigma$.

Lemma 5.14 (Feasible sets of simple fragments). *The feasible sets of simple fragments in infinite and finite string, respectively, are*

$$\mathcal{F}_\circ(l) = \begin{cases} \Gamma & \text{if } l = 1, \\ \bar{\Gamma} \times \mathcal{F}_\circ(l-1) & \text{if } l > 1. \end{cases}$$

$$\mathcal{F}_\circ^{\langle \ell \rangle}(l) = \begin{cases} \mathcal{F}_\circ(l) & \text{if } l < \ell, \\ \Sigma & \text{if } l = \ell = 1, \\ \bar{\Gamma} \times \mathcal{F}_\circ^{\langle \ell-1 \rangle}(\ell-1) & \text{if } l = \ell > 1. \end{cases}$$

Proof. The recurrence equations are immediately verified using the above considerations. □

Standard cleavage schemes. A standard fragment may have cleavage characters in its inner part, if they are immediately followed by a prohibition character. This is reflected in the extension of the wHMM by state 4 ($\bar{\Gamma} \cap \Pi = \Pi$) which can be reached from states 1 and 2 ($\Gamma \cap \bar{\Pi} = \Gamma$) (Figure 5.3).

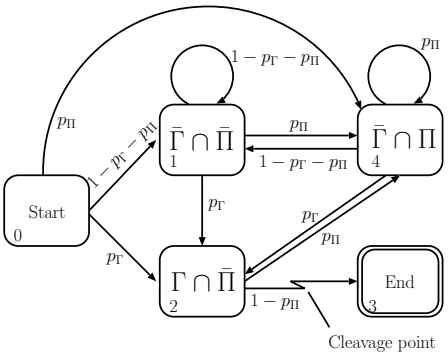

Figure 5.3.: Partly reduced weighted HMM for first fragment of standard cleavage scheme.

Note that for standard schemes, Γ and Π are pairwise disjoint, and thus Π is contained in $\bar{\Gamma}$. This allows us to join state Π with $\bar{\Gamma} \cap \Pi$ and simplify the model's transitions as in Figure 5.4 (top). We nevertheless have to keep a state Π to correctly model the output of character masses.

The first and following fragments F_1 and F_+ now have a slightly different structure: Whereas F_1 may start with any character, a following fragment F_+ cannot start with a prohibition character, since then the preceding fragment would not have ended. We solve this by introducing two different wHMMs for F_1 and F_+, as in Figure 5.4. For the F_+-wHMM, state 5 together with state 2 ensures that a following fragment does not start with a prohibition character. Note that we have to split the set $\bar{\Pi}$ into the disjoint sets $\bar{\Pi} \cap \bar{\Gamma}$ and Γ since a fragment might immediately start with a cleavage character.

The structure of standard fragments is much more complex than the structure of simple fragments; we are no longer able to give the feasible sets in explicit form but have to rely on recurrent descriptions. We also have to give these descriptions for two different fragment sets for first and following fragments, $\mathcal{F}_1$ and $\mathcal{F}_+$.

Lemma 5.15 (Feasible sets of standard fragments). *For a standard cleavage scheme* (Γ, Π)*, the sets of feasible fragment strings are given by*

$$\mathcal{F}_1(l) = \begin{cases} \Gamma & \text{if } l = 1, \\ \big(\Gamma \times \Pi \times \mathcal{F}_1(l-2)\big) \cup \big(\bar{\Gamma} \times \mathcal{F}_1(l-1)\big) & \text{if } l > 1. \end{cases}$$

$$\mathcal{F}_1^{\langle \ell \rangle}(1) = \begin{cases} \mathcal{F}_1(l) & \text{if } l < \ell, \\ \Sigma & \text{if } l = \ell = 1, \\ \big(\Gamma \times \Pi \times \mathcal{F}_1^{\langle \ell-2 \rangle}(\ell-2)\big) \cup \big(\bar{\Gamma} \times \mathcal{F}_1^{\langle \ell-1 \rangle}(\ell-1)\big) & \text{if } l = \ell > 1. \end{cases}$$

$$\mathcal{F}_+(l) = \begin{cases} \Gamma \cap \bar{\Pi} & \text{if } l = 1, \\ \big(\Gamma \times \Pi \times \mathcal{F}_1(l-2)\big) \cup \big(\bar{\Gamma} \cap \bar{\Pi} \times \mathcal{F}_1(l-1)\big) & \text{if } l > 1. \end{cases}$$

$$\mathcal{F}_+^{\langle \ell \rangle}(1) = \begin{cases} \mathcal{F}_+(l) & \text{if } l < \ell, \\ \bar{\Pi} & \text{if } l = \ell = 1, \\ \big(\big(\Gamma \cap \bar{\Pi}\big) \times \Pi \times \mathcal{F}_1^{\langle \ell-2 \rangle}(\ell-2)\big) \cup \big(\bar{\Gamma} \cap \bar{\Pi} \times \mathcal{F}_1^{\langle \ell-1 \rangle}(\ell-1)\big) & \text{if } l = \ell > 1. \end{cases}$$

Proof. The lemma is a special case of the corresponding Lemma 5.16 for general cleavage schemes. □

Interestingly, the feasible sets $\mathcal{F}_+$ for following fragments can be described in terms of $\mathcal{F}_1$ for both infinite and finite remaining string length.

There is a potential difficulty in the understanding of feasible sets for standard (and, as we will see, also general) fragments: If a substring of length l of S is a valid fragment, it is contained in $\mathcal{F}_\circ(l)$. However, the converse is not true: The first character after the substring must also not be a prohibition character. This fact is included in the wHMM models by the transition into a final state.

General cleavage schemes. Fragments of general cleavage schemes may also contain cleavage characters in their inner parts. Moreover, cleavage characters may now also be prohibition characters, introducing a lot more transitions into the wHMMs. Weighted HMMs for first and following fragments of general cleavage schemes are shown in Figure 5.5. The four states 1,2,4,5 form a partition of the weighted alphabet Σ into all combinations of cleavage/prohibition characters.

For better readability, transition probabilities are not shown. The probability to transit to some state i from any state j is the probability of the corresponding character set Σ_i: $P_{ij} = \mathbb{P}(C \in \Sigma_i)$, except for following fragments, where we already know that the first character is not a prohibition character. In this case, we have to take conditional probabilities in the transitions, i.e. $P_{01} = c_1 = \mathbb{P}(C \in \bar{\Gamma}, C \in \bar{\Pi} \mid C \in \bar{\Pi}) = \mathbb{P}(C \in \bar{\Gamma} \mid C \in \bar{\Pi})$ and $P_{02} = c_2 = \mathbb{P}(C \in \Gamma \mid C \in \bar{\Pi})$. The transition probabilities to a final state are $P_{i,End} = \mathbb{P}(C \in \bar{\Pi})$, for $i = 2, 5$, to model the non-prohibition character that completes the cleavage pattern. The two wHMMs differ in two points: The transition probabilities from the start state are different, and we are not allowed to transit from the start state to a prohibition character state 4,5 in the F_+-wHMM.

Before we formally derive the feasible sets of general fragments, let us take a glimpse at Figure 5.6, where fragments of length $l = 1, 2, 3$ are given in form of a tree-like description of the feasible sets. The figure will help us to visualize the following considerations. Note that the $\bar{\Pi}$-sets in the right do not belong to the fragment.

Let us see how we can recursively describe these fragments: Each fragment either starts with a cleavage character, or it does not. For following fragments, we have the additional constraint that a fragment must not start with a prohibition character. If a fragment does not start with a cleavage character, we are in the same situation as before and may append either a cleavage character or a non-cleavage character, that is, any character. For $L_\circ = 3$, this situation refers to the subtrees labeled (3) and (4). These two subtrees, starting in the second character, are exactly the feasible set of a first fragment of length $L_1 = 2$. What with the case that we start with a cleavage character? In that case, except for $L_\circ = 1$, the next character must be a prohibition character to thwart ending of the fragment. For general fragments, we have again to distinguish whether we append a prohibition character that is also a cleavage character (subtree (1)) or a prohibition character that is a non-cleavage character (subtree (2)). Since we are dealing with an underlying i.i.d. string model, we are now again in the same situation as before. The subtree ((1),(2)) of feasible strings starting with a prohibition character can clearly not be described by either $\mathcal{F}_1$ or $\mathcal{F}_+$, as the first starts with any character and the latter with a non-prohibition character. We therefore introduce a new set of feasible fragment suffixes (not whole fragments) of length l, denoted $\mathcal{G}(l)$. Its structure is similar to the structure of $\mathcal{F}_1(l)$ except for the first character, which must be a prohibition character. For $L_\circ = 3$, the set $\mathcal{G}(2)$ describes the subtree ((1),(2)), starting in the second character; it is the same for both F_1 and F_+.

The initial conditions are easily derived: A first fragment of length $l = 1$ may be any character, a following fragment any non-prohibition character, and the suffix set $\mathcal{G}(1)$ is the set of prohibition characters.

Similarly, we derive a recurrence for the feasible sets of fragments in finite strings with given remaining string length ℓ by adjusting the initial conditions in the case $l = \ell$.

We thus proved the following lemma which states these considerations in a more formal way.

Lemma 5.16 (Feasible sets of general fragments). *For a general cleavage scheme* (Γ, Π)*, the feasible sets of first fragments are given by*

$$\mathcal{F}_1(l) = \begin{cases} \Gamma & \text{if } l = 1, \\ \big(\bar{\Gamma} \times \mathcal{F}_1(l-1)\big) \cup \big(\Gamma \times \mathcal{G}(l-1)\big) & \text{if } l > 1. \end{cases}$$

$$\mathcal{F}_1^{\langle \ell \rangle}(l) = \begin{cases} \mathcal{F}_1(l) & \text{if } l < \ell, \\ \Sigma & \text{if } l = \ell = 1, \\ \big(\bar{\Gamma} \times \mathcal{F}_1^{\langle \ell-1 \rangle}(\ell-1)\big) \cup \big(\Gamma \times \mathcal{G}^{\langle \ell-1 \rangle}(\ell-1)\big) & \text{if } l = \ell > 1. \end{cases}$$

For following fragments

$$\mathcal{F}_+(l) = \begin{cases} \Gamma \cap \bar{\Pi} & \text{if } l = 1, \\ \big(\bar{\Gamma} \cap \bar{\Pi} \times \mathcal{F}_1(l-1)\big) \cup \big(\Gamma \cap \bar{\Pi} \times \mathcal{G}(l-1)\big) & \text{if } l > 1. \end{cases}$$

$$\mathcal{F}_+^{\langle \ell \rangle}(l) = \begin{cases} \mathcal{F}_+(l) & \text{if } l < \ell, \\ \bar{\Pi} & \text{if } l = \ell = 1, \\ \big(\bar{\Gamma} \cap \bar{\Pi} \times \mathcal{F}_1^{\langle \ell-1 \rangle}(\ell-1)\big) \cup \big(\Gamma \times \bar{\Pi} \times \mathcal{G}^{\langle \ell-1 \rangle}(\ell-1)\big) & \text{if } l = \ell > 1. \end{cases}$$

The set $\mathcal{G}(l)$ *of feasible fragment suffixes of length* l *starting with a prohibition character is given by a very similar recurrence, namely*

$$\mathcal{G}(l) = \begin{cases} \Gamma \cap \Pi & \text{if } l = 1, \\ \big(\bar{\Gamma} \cap \Pi \times \mathcal{F}_1(l-1)\big) \cup \big(\Gamma \cap \Pi \times \mathcal{G}(l-1)\big) & \text{if } l > 1. \end{cases}$$

$$\mathcal{G}^{\langle \ell \rangle}(l) = \begin{cases} \mathcal{G}_1(l) & \text{if } l < \ell, \\ \Pi & \text{if } l = \ell = 1, \\ \big(\bar{\Gamma} \cap \Pi \times \mathcal{F}_1^{\langle \ell-1 \rangle}(\ell-1)\big) \cup \big(\Gamma \cap \Pi \times \mathcal{G}^{\langle \ell-1 \rangle}(\ell-1)\big) & \text{if } l = \ell > 1. \end{cases}$$

Proof. We prove the Lemma by induction, where the initial conditions are obvious, the induction step was already given above. □

Note that there is no direct correspondence to the sets $\mathcal{G}(l)$ in the wHMMs.

Lemma 5.14 and Lemma 5.15 are special cases of the previous Lemma 5.16, and so are the corresponding wHMMs and feasible sets. In particular, we have $\mathcal{G}(l) = \emptyset$ for all $l \in \mathbb{N}$ for simple schemes and $\mathcal{G}(1) = \emptyset$ for $l = 1$ and $\mathcal{G}(l) = \Pi \times \mathcal{F}_1(l-1)$ for $l > 1$ for standard schemes.

When deriving statistics on these sets in later chapters, we have to be careful to also consider the additional constraint that the character following a fragment, although not part of the fragment, must not be a prohibition character in order to complete the cleavage pattern.

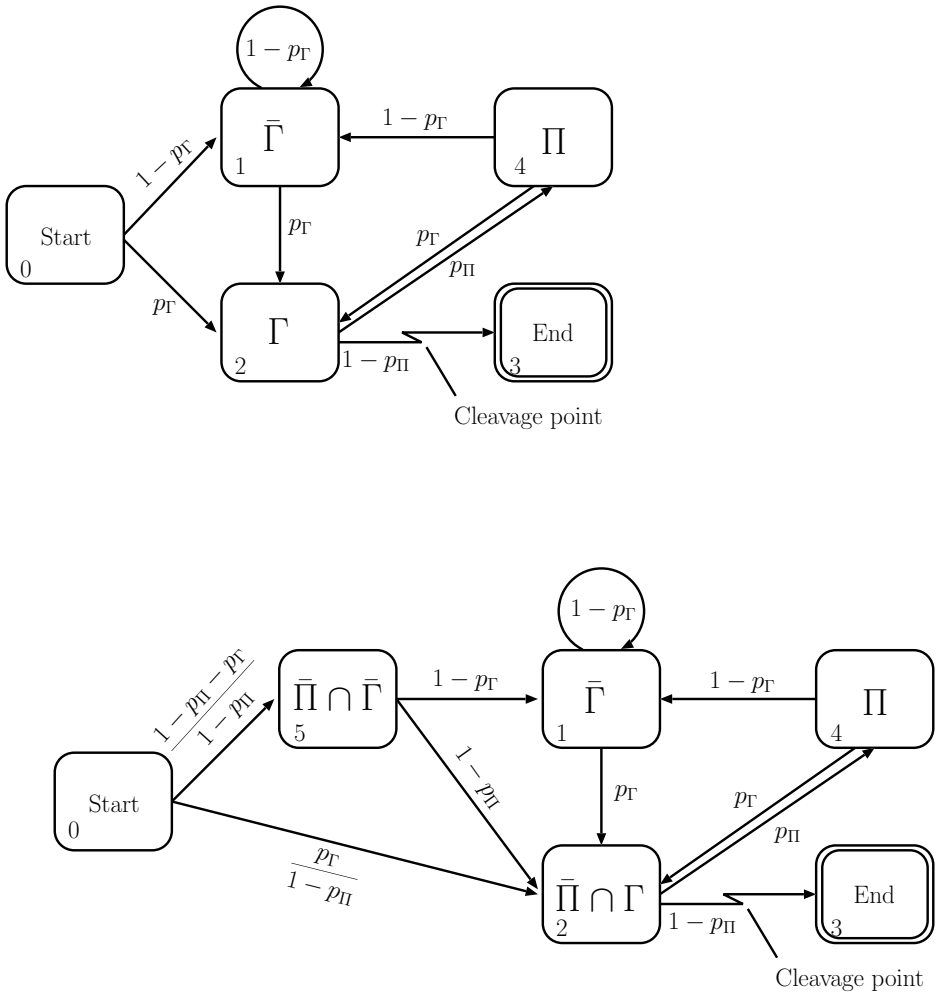

Figure 5.4.: wHMMs for the first (top) and following (bottom) fragments in a random i.i.d. string using a standard cleavage scheme (Γ, Π). Note that due to the different mass output distributions, we cannot join states 1 and 4 in either model.

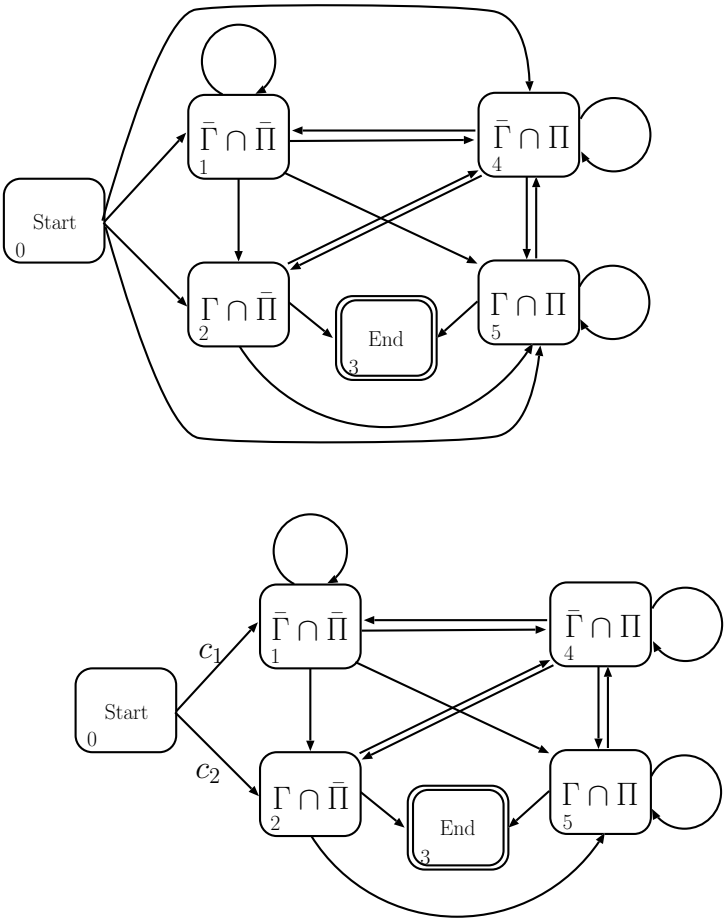

Figure 5.5.: wHMMs for general fragments, see text for details on transition probabilities. Top: First fragment. Bottom: Following fragment.

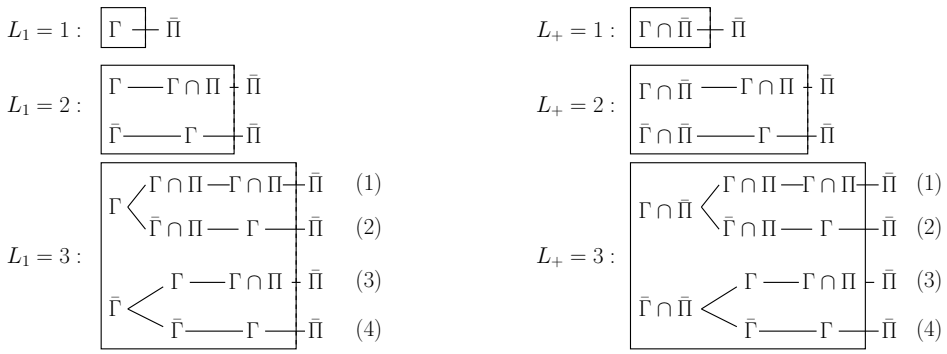

Figure 5.6.: Structure of general fragments. Left: First fragment of length $l = 1, 2, 3$. Right: Following fragment of length $l = 1, 2, 3$.

6. Distribution of Fragment Length

Let (S, μ) be an infinite i.i.d. random weighted string and let F_i be its i-th fragment of length L_i under a cleavage scheme (Γ, Π). Since the first fragment has a different prefix than the following ones, but all following ones are i.i.d., we define

$$u_1(l) := \mathbb{P}(L_1 = l),$$
$$u_+(l) := \mathbb{P}(L_+ = l) = \mathbb{P}(L_i = l) \text{ for } i \geq 2.$$

We adopt the notational conventions of Section 5.3 and write $u_\circ(\cdot)$ to denote both $u_1(\cdot)$ and $u_+(\cdot)$. The case of finite strings is covered later in this section.

6.1. Computation in wHMM Framework

In the wHMM framework, the length of a fragment is the number of steps until the wHMM arrives in a final state, where the initial step to transit from the start state and the final step for entering the final state do not contribute to the fragment length. If again W_k denotes the wHMM's state after the k-th transition, this means

$$\mathbb{P}(L_\circ = l) \;=\; \mathbb{P}(W_{l+1} \in T).$$

To compute the length distributions for general cleavage schemes, consider the wHMMs in Figure 5.5. The distributions Q_i for the mass emitted in state i are not relevant for the fragment's length and we can concentrate solely on the underlying Markov chain (E, P, p^0) together with the set of final states T. For an introduction to Markov chain theory, in particular the Chapman-Kolmogorov equation, see [45, Chapter XV].

Theorem 6.1 (Distribution of fragment length). *Given a wHMM $(E, P, p^0, T, (\Sigma, \mu), Q)$ for either a first or a following fragment, we have*

$$u_\circ(l) \;=\; \sum_{i \in T} (p^0 \cdot P^{l+1})_i,$$

the Markov chain parameters depending on the fragment type considered. It is the distribution of the Markov chain's arrival time in a final state.

Proof. Let p^l be the l-step state distribution, that is, p^l_i denotes the probability of being in state i after l steps. Then the Chapman-Kolmogorov equation from classical Markov chain theory states that $p^l = p^0 \cdot P^l$. To achieve fragment length exactly l, we need to be in a final state $i \in T$ after $l + 1$ steps, which leads to the stated formula. □

Using Markov chain theory, we can obtain closed formulas for the length distributions of both fragment types for standard cleavage schemes, using the wHMMs in Figure 5.4. The following Lemma was first given and proved in [73] by Sven Rahmann.

Lemma 6.2 (Closed formula for $u_\circ$). *Given a wHMM $(E, P, p^0, T, (\Sigma, \mu), Q)$ for either a first or a following fragment under a* standard *cleavage scheme (Figure 5.4), let*

$$\begin{aligned}\alpha &= \sqrt{(1-p_\Gamma)^2 + 4p_\Gamma p_\Pi},\\ \lambda_1 &= (1-p_\Gamma+\alpha)/2, \text{ and}\\ \lambda_2 &= (1-p_\Gamma-\alpha)/2.\end{aligned}$$

Then the length distributions $u_\circ(\cdot)$ can be written in closed form as

$$\begin{aligned}u_1(l) &= \frac{p_\Gamma(1-p_\Pi)}{\alpha}(\lambda_1^l - \lambda_2^l), \text{ and}\\ u_+(l) &= \frac{p_\Gamma}{\alpha}\cdot\Big((1-p_\Pi-p_\Gamma)(\lambda_1^{l-1}-\lambda_2^{l-1}) + p_\Gamma p_\Pi(\lambda_1^{l-2}-\lambda_2^{l-2})\Big).\end{aligned}$$

Proof. Consider the wHMM in Figure 5.4 (top). From the point of view of outgoing transitions, states 0, 1 and 4 are equivalent, so we merge them into state 1, thus obtaining a Markov chain with the following transition matrix $A = (A_{ij})$, where A_{ij} is the conditional probability of moving to state j, being in state i:

$$A = \begin{pmatrix} 1-p_\Gamma & p_\Gamma & 0\\ p_\Pi & 0 & 1-p_\Pi\\ 0 & 0 & 0\end{pmatrix}.$$

Let p_i^l denote the probability of being in state i after l steps, and let $p^l := (p_i^l)_{i=1,2,3}$. Then because of the start state now being state 1, $p^0 = (1,0,0)$, and $p^{l+1} = p^l \cdot A$, so $p^l = p^0 \cdot A^l$. Since cleavage occurs *before* entering state 3, $u_1(l) = p_3^{l+1} = (p^0 \cdot A^{l+1})_3 = A_{1,3}^{l+1}$. We obtain an explicit representation of the powers of A by diagonalization: $A = B\Lambda B^{-1}$ with an invertible matrix B and a diagonal matrix Λ so $A^l = (B\Lambda B^{-1})^l = B\Lambda^l B^{-1}$. Using the quantities α, λ_1, λ_2 as stated in the lemma, it is straightforward to verify that

$$A = \begin{pmatrix} \frac{\lambda_1}{p_\Pi} & \frac{\lambda_2}{p_\Pi} & \frac{-(1-p_\Pi)}{p_\Pi}\\ 1 & 1 & \frac{(1-p_\Gamma)(1-p_\Pi)}{p_\Pi p_\Gamma}\\ 0 & 0 & 1\end{pmatrix}\cdot\begin{pmatrix}\lambda_1 & 0 & 0\\ 0 & \lambda_2 & 0\\ 0 & 0 & 0\end{pmatrix}\cdot\begin{pmatrix}\frac{p_\Pi}{\alpha} & \frac{-\lambda_2}{\alpha} & \frac{(1-p_\Pi)\lambda_2^2}{p_\Gamma p_\Pi \alpha}\\ \frac{-p_\Pi}{\alpha} & \frac{\lambda_1}{\alpha} & \frac{-(1-p_\Pi)\lambda_1^2}{p_\Gamma p_\Pi \alpha}\\ 0 & 0 & 1\end{pmatrix}.$$

The interested reader will find the following relationships helpful: $\lambda_1 + \lambda_2 = 1 - p_\Gamma$, $\lambda_1 - \lambda_2 = \alpha$, $\lambda_1\lambda_2 = -p_\Gamma p_\Pi$, and $\lambda_i^2 = (1-p_\Gamma)\lambda_i + p_\Gamma p_\Pi$ for $i = 1, 2$. From this, we obtain

$$A^l = \frac{1}{\alpha}\cdot\begin{pmatrix}\lambda_1^{l+1}-\lambda_2^{l+1} & p_\Gamma(\lambda_1^l - \lambda_2^l) & (1-p_\Pi)p_\Gamma(\lambda_1^{l-1}-\lambda_2^{l-1})\\ p_\Pi(\lambda_1^l-\lambda_2^l) & p_\Gamma p_\Pi(\lambda_1^{l-1}-\lambda_2^{l-1}) & (1-p_\Pi)p_\Gamma p_\Pi(\lambda_1^{l-2}-\lambda_2^{l-2})\\ 0 & 0 & 0\end{pmatrix},$$

completing the proof for $u_1(\cdot)$.

The proof for $u_+(\cdot)$ is similar, considering the wHMM in Figure 5.4 (bottom), merging states 1, 4 and 5 into state 1, and removing state 0 by noting that $p^1 = ((1-p_\Pi-p_\Gamma)/(1-p_\Pi), p_\Gamma/(1-p_\Pi), 0)$. So

$$\begin{aligned} u_+(l) &= (p^1 \cdot A^l)_3 = (1-p_\Pi-p_\Gamma)/(1-p_\Pi) \cdot A_{1,3}^{l-1} + p_\Gamma/(1-p_\Pi) \cdot A_{2,3}^{l-1} \\ &= \frac{p_\Gamma}{\alpha} \cdot \Big((1-p_\Pi-p_\Gamma)(\lambda_1^{l-1} - \lambda_2^{l-1}) + p_\Gamma p_\Pi (\lambda_1^{l-2} - \lambda_2^{l-2}) \Big), \end{aligned}$$

giving the stated result. □

The same procedure of diagonalization of the transition matrix might also be possible for general fragment wHMMs, although a 6×6 matrix might not be diagonalizable by analytical means. We skip this at this point and refer the reader to Section 6.2, where we derive closed forms for the length distributions by solving the corresponding recurrence equations.

If p_Π is small in comparison to p_Γ, then λ_1 becomes dominant with $\lambda_1 \approx 1 - p_\Gamma$ and $\lambda_2 \approx 0$. For the limiting case of simple cleavage schemes, $\lambda_1 = 1 - p_\Gamma$ and $\lambda_2 = 0$.

Example 6.3 (Parameters for TryptSwissProt). For tryptic digestion and the Swiss-Prot frequencies, we have

$$\begin{aligned} p_\Gamma &= 0.1125, \\ p_\Pi &= 0.0483, \\ \alpha &= 0.8996617, \\ \lambda_1 &= 0.8935809, \\ \lambda_2 &= -0.006080871, \end{aligned}$$

and the powers of λ_2 quickly decrease to zero.

6.2. Recurrence Equations

Instead of using the wHMM framework directly, we can make use of the structural Lemma 5.16 to derive recurrence equations for the length distributions in the case of general cleavage schemes.

Theorem 6.4 (Recurrence for general schemes). *For general cleavage schemes (Γ, Π), the length distribution $u_1(\cdot)$ for the first fragment in an infinite random weighted string is given by the recurrence*

$$u_1(l) = \begin{cases} p_\Gamma \cdot p_{\bar{\Pi}}, & \text{if } l = 1, \\ p_{\bar{\Gamma}} \cdot u_1(l-1) \; + \; p_\Gamma \cdot v(l-1), & \text{if } l > 1, \end{cases}$$

and the length distribution for following fragments is given by

$$u_+(l) = \begin{cases} p_{\Gamma \cap \bar{\Pi}}, & \text{if } l = 1, \\ \frac{p_{\Gamma \cap \bar{\Pi}}}{p_{\bar{\Pi}}} \cdot v(l-1) \; + \; \frac{p_{\bar{\Gamma} \cap \bar{\Pi}})}{p_{\bar{\Pi}}} \cdot u_1(l-1), & \text{if } l > 1. \end{cases}$$

The quantities $v(l)$ are given by the recurrence

$$v(l) = \begin{cases} p_{\Gamma\cap\Pi} \cdot p_{\bar{\Pi}} & \text{if } l = 1, \\ p_{\bar{\Gamma}\cap\Pi} \cdot u_1(l-1) \,+\, p_{\Gamma\cap\Pi} \cdot v(l-1) & \text{if } l > 1. \end{cases}$$

Proof. We use Lemma 5.16. The length distributions $u_\circ(l)$ give the probabilities for the corresponding feasible sets $\mathcal{F}_\circ(l)$, and $v(l)$ describes the probabilities of a string belonging to the feasible fragment suffix set $\mathcal{G}(l)$. Since we only consider i.i.d. string models, the character sets directly translate into the corresponding character probabilities, i.e. we write $\mathbb{P}(C \in \Theta)$ whenever a character set Θ occurs in the recurrence for feasible sets.

We have to multiply by $p_{\bar{\Pi}}$ to capture the following non-prohibition character after the fragment. □

The probabilities $v(\cdot)$ form a defected length distribution of all fragment-like random weighted strings (contained in $\mathcal{G}(\cdot)$) that start with a prohibition character. These probabilities sum to $\sum_{l\in\mathbb{N}} v(l) = p_\Pi$.

The recurrences for the general case are considerably more concise for standard and simple cleavage schemes. Then, $p_{\Gamma\cap\Pi} = 0$, $p_{\Gamma\cap\bar{\Pi}} = p_\Gamma$, $p_{\bar{\Gamma}\cap\Pi} = p_\Pi$ and $p_{\bar{\Gamma}\cap\bar{\Pi}} = 1-p_\Gamma-p_\Pi$, and $v(1) = 0$ and $v(l) = p_\Pi \cdot u_1(l-1)$ for $l > 1$.

Corollary 6.5 (Recurrence for standard schemes). *The length distributions $u_\circ(\cdot)$ of the first and following fragments in a random weighted string with a standard cleavage scheme can be computed recursively by the recurrence equations*

$$u_\circ(l) \;=\; p_\Gamma p_\Pi \cdot u_\circ(l-2) \,+\, (1-p_\Gamma) \cdot u_\circ(l-1),$$

with boundary conditions $u_\circ(0) = 0$, $u_1(1) = p_\Gamma(1-p_\Pi)$ and $u_+(1) = p_\Gamma$, $u_+(2) = (1-p_\Gamma-p_\Pi)p_\Gamma$.

Corollary 6.6 (Recurrence for simple schemes). *For simple schemes, the closed formulas reduce to the same term:*

$$u_\circ(l) \;=\; p_\Gamma \cdot (1-p_\Gamma)^{l-1},$$

the probability mass function of a geometric distribution with parameter p_Γ defined on $\{1, 2, \dots\}$. It describes the waiting time for the first cleavage character in a random i.i.d. string.

The recurrence equations of Theorem 6.4 for the first fragment length distribution are a system of two linear recurrence equations with constant coefficients. Solving this system gives us the length distributions for general cleavage schemes in explicit, closed form. We proceed as follows: We first derive the ordinary generating functions (OGFs) $U_\circ(z) := \sum_{l\geq 0} u_\circ(l) \cdot z^l$ for the length distributions and similar $V(z)$ as generating function for the sequence $v(l)$. We then derive the coefficients of these OGFs in closed form by identification with OGFs of known sequences. For a general introduction of how to solve recurrences using generating functions, see e.g. [119] or [80].

Lemma 6.7 (Generating functions for length distributions). *With the notations of Theorem 6.4, the ordinary generating functions $U_\circ(z)$ and $V(z)$ for the probability sequences $u_\circ(\cdot)$ and $v(\cdot)$, respectively, are*

$$U_1(z) = \frac{p_\Gamma z V(z) \,+\, p_\Gamma p_{\bar\Pi} z}{1 - p_{\bar\Gamma} z} \tag{6.1}$$

$$= \frac{p_\Gamma p_{\bar\Pi} z}{1 - (p_{\Gamma\cap\Pi} + p_{\bar\Gamma})z + (p_{\Gamma\cap\Pi} - p_\Gamma p_\Pi)z^2} \tag{6.2}$$

$$U_+(z) = \frac{p_{\bar\Gamma\cap\bar\Pi}}{p_{\bar\Pi}} \cdot z \cdot U_1(z) \,+\, \frac{p_{\Gamma\cap\bar\Pi}}{p_{\bar\Pi}} \cdot z \cdot V(z) \,+\, p_{\Gamma\cap\bar\Pi} \cdot z \tag{6.3}$$

$$V(z) = \frac{p_{\bar\Gamma\cap\Pi} z U_1(z) \,+\, p_{\Gamma\cap\Pi} p_{\bar\Pi} z}{1 - p_{\Gamma\cap\Pi} z} \tag{6.4}$$

$$= \frac{p_{\Gamma\cap\Pi} p_{\bar\Pi} z + p_{\bar\Pi}(p_\Gamma p_\Pi - p_{\Gamma\cap\Pi})z}{1 - (p_{\Gamma\cap\Pi} + p_{\bar\Gamma})z + (p_{\Gamma\cap\Pi} - p_\Gamma p_\Pi)z^2} \tag{6.5}$$

Proof. We start by multiplying the recurrence equations for the first fragment (given in Theorem 6.4) by z^l and summing from $l = 2$ to infinity:

$$\sum_{l\geq 2} u_1(l) z^l = p_{\bar\Gamma} \sum_{l\geq 2} u_1(l-1) z^l \,+\, p_\Gamma \sum_{l\geq 2} v(l-1) z^l.$$

We now correct for the term at $l = 1$ in the left, note that $u_1(0) = v(0) = 0$ and extract one z in the two sums on the right:

$$\sum_{l\geq 0} u_1(l) z^l - p_\Gamma p_{\bar\Pi} z = p_{\bar\Gamma} z \sum_{l\geq 0} u_1(l) z^l \,+\, p_\Gamma z \sum_{l\geq 0} v(l) z^l$$

$$U_1(z) - p_{\bar\Gamma} z U_1(z) = p_\Gamma z V(z) + p_\Gamma p_{\bar\Pi} z$$

$$U_1(z) = \frac{p_\Gamma z V(z) \,+\, p_\Gamma p_{\bar\Pi} z}{1 - p_{\bar\Gamma} z}.$$

Similarly, we derive

$$V(z) = \frac{p_{\bar\Gamma\cap\Pi} z U_1(z) \,+\, p_{\Gamma\cap\Pi} p_{\bar\Pi} z}{1 - p_{\Gamma\cap\Pi} z}$$

Inserting the two equations into each other and rearranging terms yields the stated result.

The OGF $U_+(z)$ is derived by using the recurrence for $u_+(\cdot)$ of Theorem 6.4, adjusting by z for the left-shift in the right-hand side of the recurrence and taking care of the case $u_+(1) = p_{\Gamma\cap\bar\Pi}$. □

Note that $U_1(1) = U_+(1) = 1$, because they are OGFs of probability distributions. Further, $V(1) = p_\Pi$; it is the OGF of a defected probability distribution.

Extracting the l-th coefficient of the corresponding OGF yields a closed form expression for $u_\circ(l) = [z^l]\, U_\circ(z)$, which also generalizes the previous closed form expression of Lemma 6.2.

Lemma 6.8 (Closed formula for $\mathbf{u}_\circ$). *Recall the notations of Theorem 6.4. Further, let*

$$\begin{aligned}
\kappa_1 &:= \frac{p_{\Gamma\cap\Pi} + p_{\bar\Gamma}}{2} + \sqrt{\left(\frac{p_{\Gamma\cap\Pi} + p_{\bar\Gamma}}{2}\right)^2 - p_{\bar\Gamma}p_{\Gamma\cap\Pi} + p_\Gamma p_{\bar\Gamma\cap\Pi}} \\
\kappa_2 &:= \frac{p_{\Gamma\cap\Pi} + p_{\bar\Gamma}}{2} - \sqrt{\left(\frac{p_{\Gamma\cap\Pi} + p_{\bar\Gamma}}{2}\right)^2 - p_{\bar\Gamma}p_{\Gamma\cap\Pi} + p_\Gamma p_{\bar\Gamma\cap\Pi}} \\
\zeta_1 &:= \frac{p_\Gamma p_{\bar\Pi}}{\kappa_1 - \kappa_2} \\
\zeta_2 &:= \frac{p_{\bar\Gamma\cap\bar\Pi}p_\Gamma - p_{\Gamma\cap\bar\Pi}p_{\Gamma\cap\Pi}}{\kappa_1 - \kappa_2} \\
\zeta_3 &:= \frac{p_{\Gamma\cap\bar\Pi}\left(p_{\bar\Gamma\cap\Pi}p_\Gamma + p_{\Gamma\cap\Pi}p_{\bar\Gamma}\right)}{\kappa_1 - \kappa_2}.
\end{aligned}$$

Then, the length distributions for general fragments are given by

$$\begin{aligned}
u_1(l) &= \zeta_1 \cdot \left(\kappa_1^l - \kappa_2^l\right) \\
u_+(l) &= \zeta_2 \cdot \left(\kappa_1^{l-1} - \kappa_2^{l-1}\right) + \zeta_3 \cdot \left(\kappa_1^{l-2} - \kappa_2^{l-2}\right) \\
&= \frac{\zeta_2}{\zeta_1} \cdot u_1(l-1) + \frac{\zeta_3}{\zeta_1} \cdot u_1(l-2)
\end{aligned}$$

Proof. Recall the generating function $U_1(z)$ of Lemma 6.7. With the abbreviations $a := p_\Gamma p_{\bar\Pi}$, $b_1 := p_{\Gamma\cap\Pi} + p_{\bar\Gamma}$ and $b_2 := p_{\Gamma\cap\Pi} - p_\Gamma p_\Pi$, the OGF has the form

$$U_1(z) = \frac{az}{1 - b_1 z + b_2 z^2}.$$

This is a rational function in z with a polynomial of second degree in the denominator. We expand this term into a partial fraction by first bringing the denominator into the form $(1-\kappa_1 z)(1-\kappa_2 z)$ and then computing the partial fractions' coefficients. Finding the roots of the denominator and comparing coefficients yields

$$\begin{aligned}
\kappa_1 &= \frac{b_1}{2} + \sqrt{\left(\frac{b_1}{2}\right)^2 - b_2}, \\
\kappa_2 &= \frac{b_1}{2} - \sqrt{\left(\frac{b_1}{2}\right)^2 - b_2}.
\end{aligned}$$

To compute the partial fractions, consider the equation

$$\frac{az}{(1-\kappa_1 z)(1-\kappa_2 z)} = \frac{A}{1-\kappa_1 z} + \frac{B}{1-\kappa_2 z}.$$

Expanding the two terms on the right side by $(1-\kappa_2 z)$ and $(1-\kappa_1 z)$, respectively, yields

$$A + B - (A\kappa_2 + B\kappa_1)z = az,$$

from which we derive the equation system

$$\begin{aligned} A + B &= 0 \\ A\kappa_2 + B\kappa_1 &= -a. \end{aligned}$$

Solving this system for A and B yields

$$U_1(z) \;=\; \frac{a}{\kappa_1 - \kappa_2} \cdot \frac{1}{1 - \kappa_1 z} \;-\; \frac{a}{\kappa_1 - \kappa_2} \cdot \frac{1}{1 - \kappa_2 z}.$$

The two fractions containing the variable z are known OGFs for which a closed form is readily available: For any OGF $G(z) := \sum_{l \geq 0} g_l z^l$ with closed form

$$G(z) \;=\; \frac{1}{(1 - \lambda z)},$$

the coefficients are

$$g_l \;=\; \left[z^l\right] \left(\frac{1}{1 - \lambda z}\right) \;=\; \lambda^l,$$

see the previously mentioned literature for a proof. This concludes the derivation of the coefficients of $U_1(z)$.

In the exact same way, we first derive a closed form expression for $v(l) = \left[z^l\right] V(z)$, namely by a partial fraction expansion of

$$V(z) \;=\; z \cdot \frac{d_1 + d_2 z}{1 - b_1 z + b_2 z^2},$$

yielding

$$v(l) \;=\; \frac{d_1}{\kappa_1 - \kappa_2} \cdot \left(\kappa_1^l \;-\; \kappa_2^l\right) \;+\; \frac{d_2}{\kappa_1 - \kappa_2} \cdot \left(\kappa_1^{l-1} \;-\; \kappa_2^{l-1}\right).$$

Using the closed form expression for $u_1(\cdot)$ and $v(\cdot)$ finally yields the stated result for $u_+(\cdot)$. □

For standard cleavage schemes, the previous closed form expressions coincide with the previously derived expression of Lemma 6.2. In this case, we have $\kappa_i = \lambda_i$ and similarly for the ζ_i coefficients. For simple schemes, $\kappa_1 = p_{\bar{\Gamma}}$, $\kappa_2 = 0$ and $\zeta_1 = p_\Gamma / p_{\bar{\Gamma}}$, $\zeta_2 = p_\Gamma$ and finally $\zeta_3 = 0$.

6.3. Moments

Having derived the generating functions of the length distributions gives us another advantage: We can now compute moments for these distributions under a general cleavage scheme and get information about the average length of fragments. These moments will also be useful for computing the correct parameters for approximating the length distributions by appropriate geometric distributions in Section 6.4 and for approximating the cleavage point distributions in Section 7.2.

We use the generating functions $U_\circ(z)$ together with the identities

$$\mathbb{E}(L_\circ) = \left.\frac{dU_\circ(z)}{dz}\right|_{z=1}$$

to derive the expectations from the derivatives of $U_\circ(z)$ evaluated at $z = 1$. We restrict ourselves to the computation of the expected length of fragments for general cleavage schemes, noting that by more elaborate computations, we are also able to derive the variance and higher moments from the generating functions.

Lemma 6.9 (Expected length of general fragments). *Let L_* denote the length of a fragment suffix starting with a prohibition character, so $v(l) = \mathbb{P}(L_* = l)$.*

With the notations of Theorem 6.4, the expected lengths of general fragments are

$$\begin{aligned}
\mathbb{E}(L_1) &= \frac{1 - p_{\Gamma\cap\Pi} + p_\Gamma p_\Pi}{p_\Gamma p_{\bar\Pi}} \\
\mathbb{E}(L_+) &= \frac{p_{\bar\Gamma\cap\bar\Pi}}{p_{\bar\Pi}} \cdot (\mathbb{E}(L_1) + 1) \;+\; \frac{p_{\Gamma\cap\bar\Pi}}{p_{\bar\Pi}} \cdot (\mathbb{E}(L_*) + p_\Pi) \;+\; p_{\Gamma\cap\bar\Pi} \\
\mathbb{E}(L_*) &= \frac{p_{\bar\Gamma\cap\Pi} \cdot \big(1 + \mathbb{E}(L_1) + p_{\Gamma\cap\Pi} \cdot (1 - \mathbb{E}(L_1))\big) \;+\; p_{\Gamma\cap\Pi} p_{\bar\Pi}}{(1 - p_{\Gamma\cap\Pi})^2}
\end{aligned}$$

Proof. To derive the expectation $\mathbb{E}(L_1)$, we compute the first derivative of $U_1(z)$ using equation (6.2) of Lemma 6.7 and evaluate it at $z = 1$. We may then compute the expectation $\mathbb{E}(L_*)$ by using equation (6.4) and finally the expectation $\mathbb{E}(L_+)$ by using equation (6.3). Note that $U_1(1) = U_+(1) = 1$ (they are probability generating functions) and $V(1) = p_\Pi$. □

The expected length of the first and following fragments are more concise for standard and simple schemes.

Corollary 6.10 (Expected length of standard fragments). *For standard cleavage schemes, the expected fragment lengths are*

$$\begin{aligned}
\mathbb{E}(L_1) &= \frac{1}{p_\Gamma(1 - p_\Pi)} + \frac{p_\Pi}{1 - p_\Pi}, \\
\mathbb{E}(L_+) &= \frac{1}{p_\Gamma(1 - p_\Pi)}.
\end{aligned}$$

The expectations for the two standard fragment types are the same except for an additional correction term for the first fragment to compensate for the non-prohibition character after the fragment to complete the cleavage pattern. This correction term is not present for the following fragments since we condition on the first character not to be a prohibition character. The function $f(p_\Pi) = p_\Pi/(1 - p_\Pi)$ increases exponentially for $p_\Pi \in (0, 1) \subset \mathbb{R}$:

p_Π	0	0.1	0.2	0.3	0.5	0.8	0.9	0.99	1
$f(p_\Pi)$	0	$0.111\cdots$	0.25	0.4	1	4	9	99	∞

.

We may conclude that the two fragment types for standard cleavage schemes have comparable expected length if prohibition characters are not dominantly frequent. Even for $p_\Pi = 0.9$ and $p_\Gamma = 0.1$, the two expected lengths are $\mathbb{E}(L_1) = 109$ and $\mathbb{E}(L_+) = 100$, respectively, so the first fragment is only slightly longer on average.

Corollary 6.11 (Expected length of simple fragments). *For simple cleavage schemes, Corollary 6.10 applies with $p_\Pi = 0$ and thus*

$$\mathbb{E}(L_\circ) \;=\; \frac{1}{p_\Gamma},$$

the expectation of a geometric random variable with parameter p_Γ.

Example 6.12 (Moments for TryptSwissProt). For Swiss-Prot frequencies and tryptic digestion, we computed the expectation and standard deviation of the fragment length and estimated their empirical counterparts from the Swiss-Prot database:

	Model	Estimate
$\mathbb{E}(L_1)$	9.39	10.18
$\mathrm{sd}(L_1)$	8.88	10.70
$\mathbb{E}(L_+)$	9.34	9.01
$\mathrm{sd}(L_+)$	8.88	9.55

indicating that the two distributions are very similar, but nevertheless different. The standard deviations were computed using the identities $\mathrm{sd}(L_\circ) = \sqrt{\mathrm{Var}(L_\circ)}$ and

$$\mathrm{Var}(L_\circ) \;=\; \left.\frac{dU_\circ(z)}{dz}\right|_{z=1} + \left.\frac{d^2U_\circ(z)}{dz^2}\right|_{z=1} - \left(\left.\frac{dU_\circ(z)}{dz}\right|_{z=1}\right)^2.$$

6.4. Approximation

Intuitively, the two fragment length distributions should be related to geometric distributions, as they describe the waiting time for the first occurrence of a cleavage pattern. Ignoring possible self-overlaps, such a pattern has probability $p_\Gamma(1 - p_\Pi)$, which would be the parameter of the geometric distribution. Further, if a random variable X has geometric distribution $\mathcal{L}(X) = \mathrm{Geom}(p)$, its parameter p can be computed as $p = 1/\mathbb{E}(X)$ if the expectation is known. For the setting TryptSwissProt, we get

$$\begin{aligned} p_\Gamma(1 - p_\Pi) &= 0.1070663, \\ 1/\mathbb{E}(L_1) &= 0.1064876, \\ 1/\mathbb{E}(L_+) &= 0.1070663, \end{aligned}$$

so the choice does not make much of a difference. Indeed, the expected length of a following fragment is exactly the expected length of a geometric distribution. For non-standard cleavage schemes, self-overlaps in the cleavage patterns become more important and the geometric approximation may not be as good as for standard schemes.

A comparison between the exact length distributions and the approximating $\mathrm{Geom}\big(1/\mathbb{E}(L_\circ)\big)$ is shown in Figure 6.1. The quantile-quantile plots show a nearly perfect bisector, indicating a very good approximation quality.

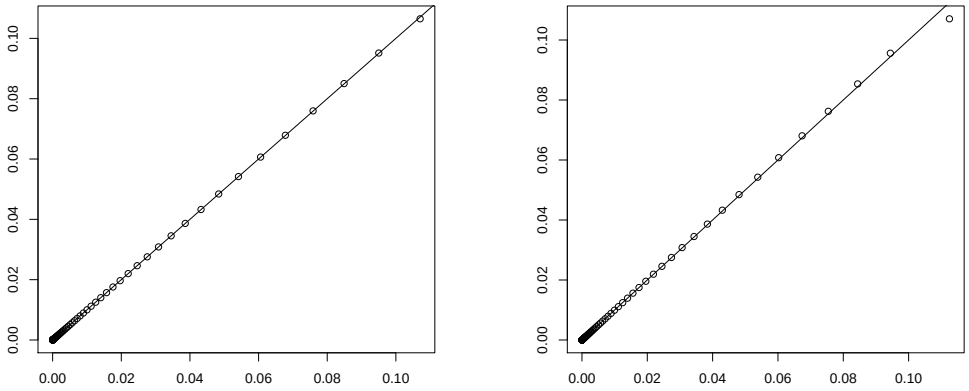

Figure 6.1.: Quantile-quantile plot of approximations of $u_1(\cdot)$ (left) and $u_+(\cdot)$ (right) by geometric distributions $\mathrm{Geom}(1/\,\mathbb{E}(L_\circ))$ using tryptic digestion. Solid lines are bisectors.

6.5. Finite Strings

For finite strings, several adjustments have to be given to capture the new boundaries at the end of the strings. We make the following definitions:

$$u_1^{\langle \ell \rangle}(l) := \mathbb{P}(L_1^{\langle \ell \rangle} = l),$$
$$u_+^{\langle \ell \rangle}(l) := \mathbb{P}(L_i^{\langle \ell + k \rangle} = l \mid C_{i-1} = k) \text{ for any } i \geq 2 \text{ and any } k \in \mathbb{N}.$$

The second definition is in fact independent of i and k, and defines the conditional fragment length distribution given that there are ℓ characters left in the string. It is *not* the length distribution of a following fragment in a finite string of length ℓ but in a finite string with *remaining suffix of length* ℓ. Both distributions also cover the case that the corresponding fragment is the last in the finite string.

The length distributions on the finite string differ from their counterparts on the infinite string only in the last possible position $l = \ell$, which now takes the remaining probability mass to give a valid probability distribution.

Lemma 6.13 (Fragment length in finite strings). *The fragment length distributions for finite string length are given by* $u_\circ^{\langle \ell \rangle}(l) = u_\circ(l)$ *if* $l < \ell$*, and the boundary*

$$u_\circ^{\langle \ell \rangle}(\ell) = \sum_{l'=\ell}^{\infty} u_\circ(l') = 1 - \sum_{l'=1}^{\ell-1} u_\circ(l').$$

Proof. If $l < \ell$, the boundary condition is irrelevant and we have $u_\circ^{\langle \ell \rangle}(l) = u_\circ(l)$. For

$l = \ell$, we have $u_1^{\langle \ell \rangle}(\ell) = \mathbb{P}(L_1^{\langle \ell \rangle} = \ell) = \mathbb{P}(L_1 \geq \ell) = \sum_{l'=\ell}^{\infty} u_1(l') = 1 - \mathbb{P}(L_1 < \ell) = 1 - \sum_{l'=1}^{\ell-1} u_1(l')$, and a similar argument holds for $u_+^{\langle \ell \rangle}(\cdot)$. □

By algebraic means, we can again derive explicit formulas for the case of finite string length.

Lemma 6.14 (Exact values for $u_1^{\langle \ell \rangle}$ and $u_+^{\langle \ell \rangle}$). *Using the same notation as in Lemma 6.2,*

$$\begin{aligned} u_1^{\langle \ell \rangle}(\ell) &= \zeta_1 \cdot \left(\frac{\kappa_1^\ell}{1-\kappa_1} - \frac{\kappa_2^\ell}{1-\kappa_2} \right), \\ u_+^{\langle \ell \rangle}(\ell) &= \zeta_2 \cdot \left(\frac{\kappa_1^{\ell-1}}{1-\kappa_1} - \frac{\kappa_2^{\ell-1}}{1-\kappa_2} \right) + \zeta_3 \left(\frac{\kappa_1^{\ell-2}}{1-\kappa_1} - \frac{\kappa_2^{\ell-2}}{1-\kappa_2} \right) \\ &= \frac{\zeta_2}{\zeta_1} \cdot u_1^{\langle \ell-1 \rangle}(\ell-1) \; + \; \frac{\zeta_3}{\zeta_1} \cdot u_1^{\langle \ell-2 \rangle}(\ell-2). \end{aligned}$$

Proof. The proof is straightforward by combining Theorem 6.4 with Lemma 6.13, and computing the geometric series by

$$\sum_{l=1}^{\ell-1} \kappa_k^l \; = \; \sum_{l=0}^{\ell-1} \kappa_k^l - 1 \; = \; \frac{1-\kappa_k^\ell}{1-\kappa_k} - 1$$

for $k = 1, 2$. □

6.6. Implementation

Implementation of the length statistics computation is straightforward: We either compute the necessary parameters κ_i, $i = 1, 2$ and ζ_i, $i = 1, 2, 3$ from the given weighted alphabet and implement the closed form of the distributions. Assuming $\mathcal{O}(1)$ time complexity for arithmetic operations including powers, each probability is computed in $\mathcal{O}(1)$. We can also compute these distributions using the recurrence equations. Storing the whole distributions up to some length $\tilde{l}$, this takes $\mathcal{O}(\tilde{l})$ time since each recurrence step takes constant time using standard dynamic programming techniques.

The space complexity is $\mathcal{O}(\tilde{l})$ if the distributions are stored in memory. For the closed form, the space complexity is $\mathcal{O}(1)$ if each probability is re-computed each time and $\mathcal{O}(\tilde{l})$ if it is computed once and stored in memory.

Note that $\tilde{l}$ can usually be chosen quite small, as the probabilities decrease quickly to zero with increasing length, see next Section 6.7, in particular Figure 6.2, left and right.

Storing both distributions $u_1(\cdot)$ and $u_+(\cdot)$ in memory using double precision with 8 bytes per entry and a maximal length of $\tilde{l} = 350$, the memory consumption is $2 \cdot 350 \cdot 8$ bytes, about 5.5 kB. Using Swiss-Prot frequencies and tryptic digestion, the probabilities at this length are $u_\circ(350) \approx 9.3 \cdot 10^{-19}$.

For computing the finite length distributions, given the length distributions for infinite strings, we may use the relation

$$u_\circ^{\langle \ell \rangle}(\ell) \;=\; u_\circ^{\langle \ell-1 \rangle}(\ell-1) - u_\circ(\ell-1),$$

obvious from Lemma 6.13 or the explicit form of Lemma 6.14. Both have complexity $\mathcal{O}(\tilde{l})$ to compute the finite length distributions up to some length $\tilde{l}$.

6.7. Evaluation on Swiss-Prot

Using our standard setting TryptSwissProt, we computed the length distributions $\mathcal{L}(L_1)$ of the first fragment and $\mathcal{L}(L_+)$ of following fragments and compared them to the empirical distributions derived from the Swiss-Prot database and again to the approximating $\mathrm{Geom}(1/\,\mathbb{E}(L_\circ))$. The agreement between model and approximation is excellent. Not surprisingly, the model does not fit the empirical data as good: Very small fragment lengths are underestimated, especially for the first fragment. As the agreement between model and data is much better for following fragments, a possible explanation would be that the i.i.d. model misses several aspects of the amino acid compositions. It seems that in nature, tryptic cleavage characters occur more often at the beginning of a protein than inside and so the composition is not as homogeneous as assumed by the i.i.d. string model. We confirmed this by comparing the occurrence of the cleavage pattern at different position within Swiss-Prot sequences.

For first fragments, single-character fragments are heavily overestimated. This is due to the methionine-prefix common to most of the database sequences: 173 350 out of 192 433 database sequences start with a methionine. See the introduction on page 3 for a biological explanation of this phenomenon. The problem can be fixed easily: We can add an additional "methionine" state to the F_1-wHMM that captures the first letter. Another solution would be to simply ignore the first letter, as it will be a methionine with probability almost one and thus does not contribute to the stochastics of the first fragment, and treat it as another mass modification like the additional H_2O. It should also be noted that the Swiss-Prot database contains sequences of many different species; if we set up a model for a certain species, it is known whether the first amino acid is a methionine or not, and we can adjust the model accordingly. In the context of mass spectrometry, this problem is even less important as the methionine prefix is removed by a post-translational modification in most organisms.

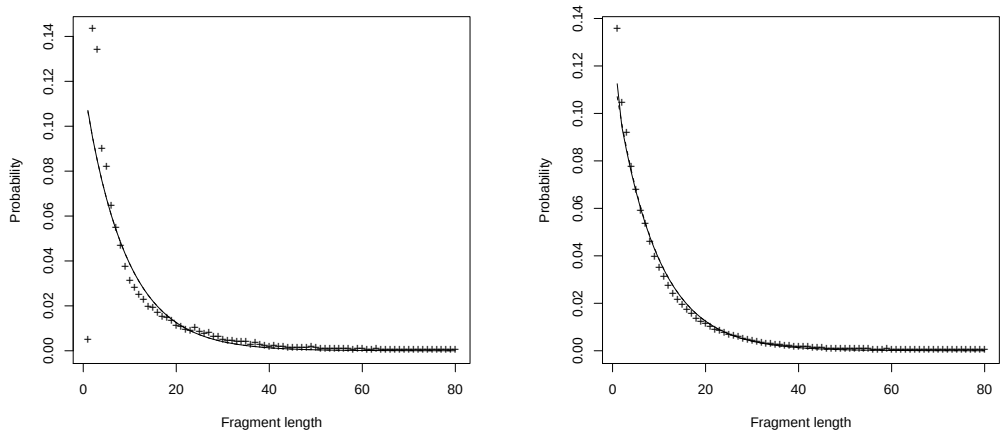

Figure 6.2.: Exact fragment length distributions and their empirical counterparts of the Swiss-Prot database using tryptic digestion. Left: Length distribution $u_1(\cdot)$ of first fragment. Right: Length distribution $u_+(\cdot)$ of following fragments. Solid lines: Exact distributions according to the model. Dashed lines: Geometric distribution with $p = 1/\mathbb{E}(L_\circ)$. Points: Empirical distributions.

7. Distribution of Cleavage Points and Fragmentation Size

Using the length distribution of fragments, we are able to compute the distribution of cleavage points in random weighted strings. With these distributions, we can also establish the distribution of the fragmentation size of finite random weighted strings.

7.1. Distribution of Cleavage Points

Since fragments do not overlap, the cleavage points can be expressed as sums of fragment lengths: The k-th cleavage point C_k is given by

$$C_k \;=\; L_1 + L_2 + L_3 + \cdots + L_k.$$

In an infinite string, the lengths of fragments are mutually independent, so the distribution of cleavage point C_k can be computed by convolution of k fragment length distributions. The following lemma is a standard result for regenerative processes.

Lemma 7.1 (Distribution of cleavage points). *For an infinite random weighted string and a general cleavage scheme, the distribution of the k-th cleavage point C_k is given by*

$$\mathcal{L}(C_k) = \mathcal{L}(L_1) \star \mathcal{L}(L_+)^{\star(k-1)} = \mathcal{L}(C_{k-1}) \star \mathcal{L}(L_k),$$

for any $k \geq 1$ and $\mathcal{L}(C_1) = \mathcal{L}(L_1)$ $(k = 1)$ in particular.

The expectation and variance of the k-th cleavage point C_k are

$$\begin{aligned}
\mathbb{E}(C_k) &= \mathbb{E}(L_1) \;+\; (k-1) \cdot \mathbb{E}(L_+), \\
\operatorname{Var}(C_k) &= \operatorname{Var}(L_1) \;+\; (k-1) \cdot \operatorname{Var}(L_+) \\
&= \mathbb{E}(L_1^2) - \mathbb{E}(L_1)^2 \;+\; (k-1) \cdot \left(\mathbb{E}(L_+^2) - \mathbb{E}(L_+)^2\right).
\end{aligned}$$

Proof. We have $C_k = \sum_{i=1}^{k} L_i = C_{k-1} + L_k$; this also holds for $k = 1$, since $C_0 = 0$ by definition. Using the independence of the L_i allows us to compute the distribution of this sum by convolution.

The moment equations follow directly from the linearity of the expectation, the equality $\operatorname{Var}(X) = \mathbb{E}(X^2) - \mathbb{E}(X)^2$ and the independence of the L_i, which gives $\mathbb{E}(L_1 L_+) = \mathbb{E}(L_1)\,\mathbb{E}(L_+)$ and $\operatorname{Var}(L_i + L_j) = \operatorname{Var}(L_i) + \operatorname{Var}(L_j)$. □

7.2. Approximation of Cleavage Point Distributions

In Section 6, we successfully established geometric distributions with parameters $1/\mathbb{E}(L_1)$ and $1/\mathbb{E}(L_+)$, respectively, as approximations for the length distributions in the case of standard cleavage schemes.

It is sensible to try to approximate the distribution of cleavage point C_k by a negative binomial distribution: A negative binomial distribution NegBin(i,p) is the distribution of the waiting time for the i-th success in a Bernoulli trial with success-probability p, not counting the $i-1$ previous successes. As such, it is the distribution of the sum of i geometric random variables each with identical parameter p, shifted by $+i$. Here, we deal with a sum of two different but very similar approximating geometric distributions for L_1 and L_+; we may take a weighted average of the two expectations to compute the parameter p. The distribution of cleavage point C_k can then be approximated by a negative binomial distribution with size parameter $i = k$ and probability parameter $p = k/\left(\mathbb{E}(L_1) + (k-1)\,\mathbb{E}(L_+)\right)$ shifted by $+k$.

7.3. Distribution of Fragmentation Size

From the cleavage point distributions, we can derive the distribution $\mathcal{L}(N^{\langle\ell\rangle})$ of fragmentation size. As we have already seen, the cleavage points form a renewal sequence on the infinite random weighted string S; therefore results from renewal theory apply. Establishing a connection between $\mathcal{L}(N^{\langle\ell\rangle})$, which is a quantity on the finite string, and the cleavage point distributions $\mathcal{L}(C_k)$, quantities on the infinite string, allows us to use these results to get the exact distribution of the cleavage points and the number of fragments.

Lemma 7.2 (Relationship of $N^{\langle\ell\rangle}$ and C_k). *The fragmentation size $N^{\langle\ell\rangle}$ of a random weighted string of length ℓ is related to the location of cleavage points by*

$$\mathbb{P}(N^{\langle\ell\rangle} \leq k) = \mathbb{P}(C_k \geq \ell).$$

Proof. If the k-th cleavage point C_k lies at ℓ or beyond, the number of fragments up to position ℓ is at most k, and vice versa. □

7.4. Evaluation on Swiss-Prot

Cleavage points. We compared the distributions of cleavage points for tryptic digestion to their empirical counterparts estimated from the empirical length distributions. Figure 7.1 shows this comparison for C_5 and C_{40}. As dashed lines, the approximating negative binomial distribution is plotted with probability parameter $p = k/(\mathbb{E}(L_1) + (k-1)\,\mathbb{E}(L_+))$ for size parameters $k = 5$ and $k = 40$, respectively. The agreement between model and approximation is again excellent as shown in Figure 7.1, upper panel and lower left. As for the empirical data, the agreement gets worse with increasing index. Whereas the model fits the empirical data for C_5 quite reasonable, the

agreement between C_{40} and its empirical counterpart is not as good; in both cases, the main aspects of the distributions are nevertheless covered. The lower part of Figure 7.1 shows quantile-quantile plots for the exact distribution of C_{40} versus the approximating negative binomial and the empirical distribution, respectively. Again, the negative binomial shows an excellent agreement; all points are on the bisector. For the empirical distribution, we see a disagreement with the model in the lower tail, where the quantiles are below the bisector. This disagreement is caused by the model's underestimation of small fragment lengths as already discussed in Section 6.7: As real fragments are shorter, the cleavage points occur earlier in the string than predicted by the model. The distribution's disagreement diminishes for higher quantiles.

Number of fragments. As a result of a short evaluation of the sequence lengths contained in the Swiss-Prot database, we found that a length about 200 lead to the maximal number of corresponding protein sequences in the database (Figure 7.2 (left)). However, there are only a few hundred protein sequences of length exactly 200 in the database. To get a more reliable estimate of the fragmentation size distribution, we pooled all sequences of length 200 up to 215, with a total of 7 050 protein sequences, and compared these to the length 207 in the model. Figure 7.2 (right) shows this comparison of the fragmentation size for proteins of length 207 together with a Gaussian approximation.

Our exact distribution underestimates the tail probabilities but nevertheless agrees better to the empirical data than the normal approximation. This is also reflected in the comparison of the first two moments: Expected value and standard deviation for the exact distribution are numerically evaluated to 22.0 ± 6.07, and for the empirical distribution 25.1 ± 7.86.

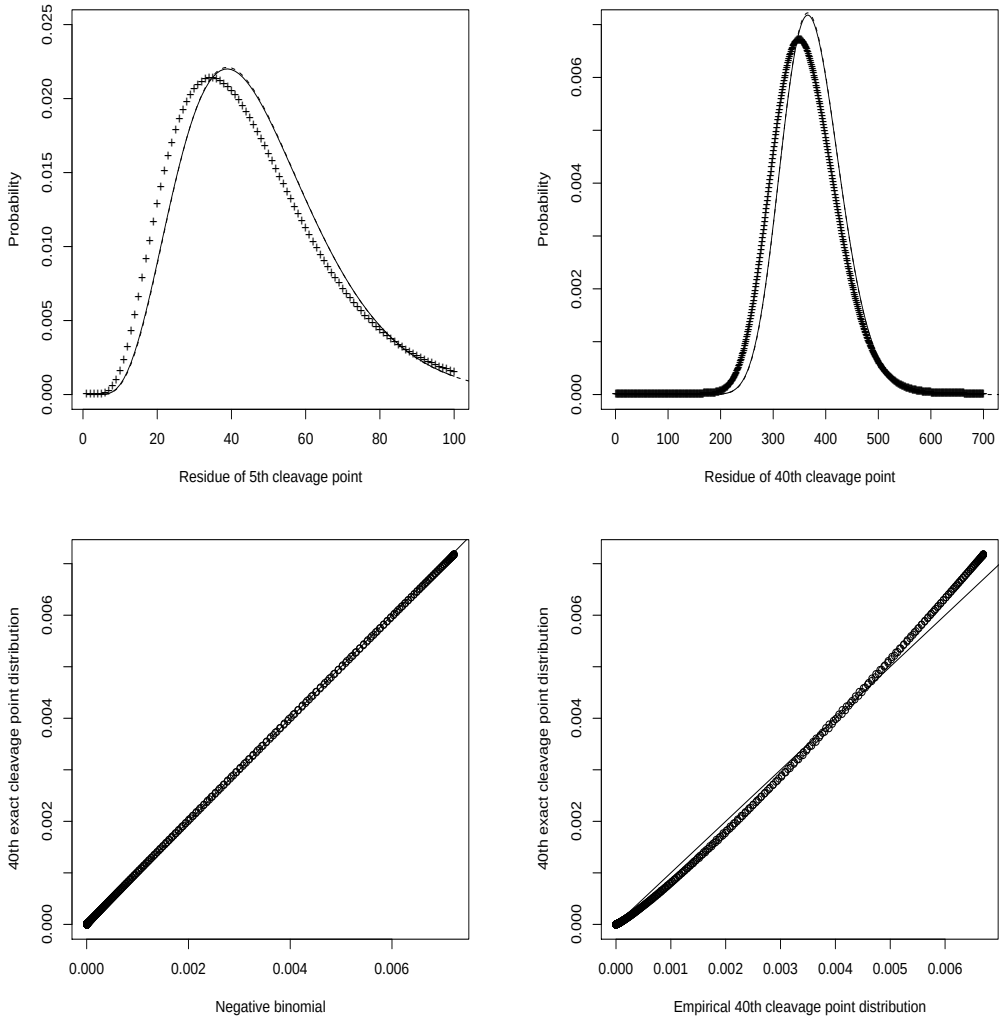

Figure 7.1.: Upper: Cleavage point distributions of C_5 (left) and C_{40} (right). Pluses: Empirical distribution derived from empirical length distributions using Swiss-Prot. Solid line: Exact theoretical distribution. Dashed line: Approximation by negative binomial (see text for parameters). Lower: Quantile-quantile plots of exact C_{40} distribution vs. negative binomial approximation (left) and empirical Swiss-Prot data (right), respectively. Solid lines are bisectors.

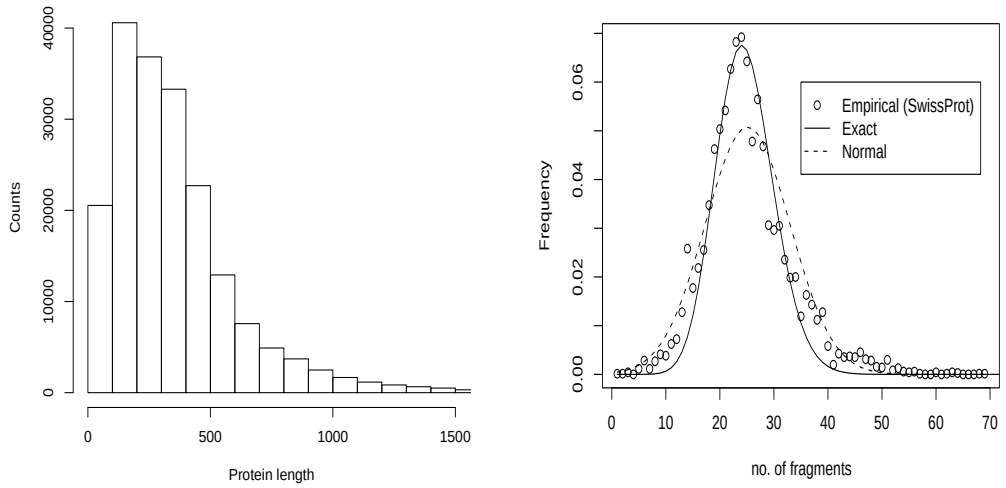

Figure 7.2.: Left: Histogram of protein lengths up to 1 500 in Swiss-Prot. Right: Distribution of fragmentation size $N^{\langle \ell \rangle}$. Points: Empirical distribution derived from Swiss-Prot for $\ell = 200 \ldots 215$. Solid line: Exact theoretical distribution for $\ell = 207$. Dashed line: Normal approximation for $\ell = 207$

8. Joint Distribution of Fragment Length and Mass

As a next step, we investigate the joint distributions of fragment length and mass. These distributions give the probability that a certain fragment has some length l and mass m. From them, we also derive related distributions such as the distribution of fragment mass; they are also of importance for computing mass occurrence probabilities in Chapter 9. As for the length distributions, we have to distinguish between first and following fragments. Again, we first examine the joint length-mass distributions in infinite random weighted strings and make adaptions to finite string length later.

Let us define

$$f_1(l,m) := \mathbb{P}(L_1 = l, \mu(F_1) = m),$$
$$f_+(l,m) := \mathbb{P}(L_+ = l, \mu(F_+) = m).$$

The distributions can be computed efficiently by either using the wHMM framework (Section 8.1) or combinatorial recurrence equations (Section 8.2). While the wHMM framework offers a more elegant model, time- and memory-efficient dynamic programming algorithms are more easily derived by extending the recurrence equations for the length distributions to the joint length-mass distributions.

8.1. Computation in wHMM Framework

So far, we only used the state sequence $(W_i)_{i\in\mathbb{N}_0}$ of a wHMM to compute the distributions of fragment lengths and cleavage points, but did not have to use the wHMM's emitted mass process $(\mu(W_i))_{i\in\mathbb{N}_0}$. We have already identified the state sequence and the output sequence of a wHMM with the string and mass process of this string, respectively.

Within the wHMM framework, the mass of a fragment is the accumulated mass process output by the wHMM: $\mu(S_{1:l}) = \sum_{i=1}^{l} \mu(S_i)$ for some l-prefix of the state sequence. The accumulated mass up to the arrival of the wHMM in a final state can now be identified with the fragment's mass $\mu(F_\circ)$.

As a formal model for computation, we introduce *Markov Additive Chains (MACs)* on wHMMs. They describe the accumulation of the mass process emitted by a wHMM.

Definition 8.1 (Markov Additive Chain (MAC)). A *Markov Additive Chain (MAC) of a wHMM* $(E, P, p^0, T, (\Sigma, \mu), Q)$ is a stochastic process M with index set $\mathbb{N}$ and taking values in $\mathbb{N}_0$. It is defined by its finite dimensional distributions for $I = \{i_1, \ldots, i_n\}$

$$\mathcal{L}(M_I) \;=\; \mathcal{L}\left(\mu(S_{i_1})\right) \star \cdots \star \mathcal{L}\left(\mu(S_{i_n})\right),$$

where $(S_i)_{i\in\mathbb{N}}$ is the character sequence and $\mu(S_i)$ its mass process emitted by the wHMM. We denote by M_k the accumulated string mass *up to* time k: $M_k := \mu(S_1 \dots S_k)$. Thus,

$$\mathbb{P}(M_k = m) \;=\; \mathbb{P}\left(\sum_{i=1}^{k} \mu(S_i) = m\right).$$

Note that M_k is different from $M_{\{k\}}$, the process M on the index set $I = \{k\}$, which is the mass of character S_k: $\mathbb{P}(M_{\{k\}} = m) = \mathbb{P}(\mu(\Sigma_{W_k}) = m) = q_k(m)$. In the following, we will use the prefix mass M_k exclusively.

For a cleavage fragment $F_\circ$ of length $L_\circ$ and mass $\mu(F_\circ)$, the MAC gives the joint length-mass distribution $f_\circ(\cdot,\cdot)$ as follows: The length $L_\circ$ is the first time the wHMM enters a final state; recall that we do not count the transitions from a start and to a final state. The fragment's mass is then $M_{L_\circ}$.

For computing these quantities, we recursively compute the masses accumulated up to step k, given the accumulated mass at step $k-1$. We introduce the probability $h_j^l(m)$ of being in state j after l steps and having accumulated mass m:

$$h_j^l(m) \;:=\; \mathbb{P}(W_l = j,\, M_l = m),$$

where $(W_k)_{k\in\mathbb{N}_0}$ is again the wHMM's state sequence.

For time $l = 0$, before making the first transition, the wHMM potentially emits a mass in its start state W_0 and so $h_j^0(m) = p_j^0 \cdot q_j(m)$. For times $l \geq 1$, we establish a recurrence relation.

Lemma 8.2 (Recurrence for $h_j^l(m)$). *Given a MAC on a wHMM for a general cleavage scheme, and using the previous notation, the probability $h_j^l(m) = \mathbb{P}(W_l = j,\, M_l = m)$ can be computed recursively by*

$$h_j^l(m) \;=\; \sum_{i\in E} \left(\left(h_i^{l-1} \cdot P_{ij}\right) \star q_j\right)(m) \;=\; \sum_{i\in E} P_{ij} \cdot \sum_{m'\in\mathbb{N}_0} h_i^{l-1}(m-m') \cdot q_j(m'),$$

for $l \geq 1$ and $j \in E$, where P_{ij} is the (i,j)-th entry in the transition matrix. The recurrence relation is completed by the initial condition

$$h_j^0(m) \;=\; p_j^0 \cdot q_j(m)$$

for $j \in E$.

Proof. For the initial condition, the probability of having mass m while being in state j after 0 steps is the emission probability $q_j(m)$ of mass m in state j multiplied with the probability p_j^0 that the wHMM starts in state j.

For the chain to be in state j after l steps while having accumulated mass m, the chain has to have accumulated mass $m - m'$ in the previous step and add mass m' in state j. Considering all previous states and their transitions to j gives the stated result. □

Note that although we sum over $\mathbb{N}_0$, the sum is of course bounded by $\mu_{\min}$ and $\mu_{\max}$, as the emission probabilities are zero for other masses.

For given l, summing $h_j^l(m)$ over all states $j \in E$ gives the mass distribution of a fragment prefix of length l:

$$\mathbb{P}(M_l = m) \;=\; \sum_{j \in E} h_j^l(m).$$

A fragment $F_\circ$ is finished if the wHMM reaches a final state; this gives the joint length-mass distribution for fragments from the prefix probabilities.

Theorem 8.3 (Computation of $f_\circ$). *For a given MAC on a wHMM, the joint distribution $f_\circ(\cdot,\cdot)$ of length and mass of a fragment $F_\circ$ of a general cleavage scheme is*

$$f_\circ(l, m) \;=\; \sum_{j \in T} h_j^{l+1}(m).$$

Proof. $h_j^l(m)$ gives the probability that the mass of a length-l fragment prefix is m while the wHMM is in state j. Once the wHMM reaches a final state $j \in T$, it stops and all transition probabilities from this state are zero. As the transition to the final state does not count towards the fragment's length, we get

$$\mathbb{P}(L_\circ = l,\, \mu(F_\circ) = m) \;=\; \mathbb{P}(W_{l+1} \in T,\, \mu(W_0 \dots W_{l+1}) = m)$$

by using the extension of the character mass function to sets of wHMM states (and their associated sub-alphabets), as given in Definition 5.10. Summing over all possible $j \in T$ gives the stated result. □

Matrix notation. Before taking a look at examples and then move on to finite strings and combinatorics, let us first introduce a matrix notation for the computation of joint length-mass distributions. This new notation gives rise to an elegant recurrence update formula for the wHMM/MAC framework.

In Definition 5.11, we have already written the family of mass emission distributions $Q_i = (q_i(m))_{m \in \mathbb{N}_0}$ as a matrix Q. With rows starting at mass $m = 0$, this matrix has size $(\mu_{\max}+1) \times |E|$. We have further introduced the transition matrix P of size $|E| \times |E|$ and the column vector of the initial state distribution p^0 of size $|E| \times 1$. Let $\underline{p^0}$ denote the $|E| \times |E|$ matrix where each column is given by p^0: $\underline{p^0} := \big(p^0 | p^0 | \cdots | p^0\big)$.

There are two more issues remaining: We have to introduce a matrix that keeps the prefix mass distributions $h_j^l(m)$ and we have to take care of correct matrix dimensions when performing convolution.

Let $H^{(l)}$ denote the matrix of prefix mass distributions after step l, that is

$$H^{(l)} \;=\; \Big(h_j^l(m)\Big)_{\substack{0 \le m \le (l+1)\,\mu_{\max}, \\ j \in E}}.$$

Of course, the mass of a fragment prefix after l steps is at least $(l-1)\,\mu_{\min}$ if $\mu(\varepsilon_s) = \mu(\varepsilon_e) = 0$ and the last step transits into a final state, so this prefix has $l-1$ characters.

For non-zero terminal masses, the minimal prefix mass is even greater. So, we could skip the first $(l-1)\,\mu_{\min}-1$ rows as they are zero, anyway. To see the formal reason to keep these rows in the definition of the matrix, let us extend the convolution operation to matrices in a column-wise manner.

Definition 8.4 (Convolution of matrices). Let $X = (X_1|X_2|\cdots|X_n)$ be a real matrix of size $x \times n$ and $Y = (Y_1|Y_2|\cdots|Y_n)$ be a real matrix of size $y \times n$. Then, define the convolution $X \star Y$ column-wise on the column vectors X_i and Y_i:

$$X \star Y \;:=\; (X_1 \star Y_1 \,|\, X_2 \star Y_2 \,|\, \cdots \,|\, X_n \star Y_n)\,.$$

The resulting matrix has size $(x + y - 1) \times n$.

As a major result of these considerations, we can now give the recurrence equation for the fragment prefix masses in a more elegant matrix form.

Lemma 8.5 (Matrix update equation). *Let the matrices Q, P and $H^{(l)}$ be defined as above. Then the matrix $H^{(l)}$ can be computed recursively by the update equation*

$$H^{(l)} \;=\; \left(H^{(l-1)} \cdot P\right) \star Q$$

with initial condition $H^{(0)} = Q \cdot \underline{p^0}$.

Proof. The initial condition is obvious from previous considerations. To compute $h_j^l(m)$, consider the recurrence equation in Lemma 8.2:

$$h_j^l(m) \;=\; \sum_{m' \in \mathbb{N}_0} \left(\sum_{i \in E} P_{ij} \cdot h_i^{l-1}(m - m') \right) \cdot q_j(m'),$$

which we identify as the (m, j)-th entry of $H^{(l)}$. □

Ignoring terminal characters, the convolution involves a matrix $H^{(l-1)}$ with $l\,\mu_{\max}+{+}1$ rows and a matrix Q with $\mu_{\max}+1$ rows. The resulting matrix $H^{(l)}$ thus has $l\,\mu_{\max}+1\,+\,\mu_{\max}+1\,-\,1\,=\,(l+1)\,\mu_{\max}+1$ rows, so each step increases the number of rows in the resulting matrix by $\mu_{\max}$.

Examples. Let us have a look at two examples for illustrating the theory: In the first example, we will use the matrix notation for a simple scheme and a very small alphabet of size 2. In the second example, we shall investigate the explicit recurrences for standard cleavage schemes.

Example 8.6 (Computation of $f_\circ$ for simple cleavage scheme). Consider a simple cleavage scheme and the associated wHMM in Figure 5.2. Further consider a terminal-extended weighted alphabet $\Sigma = \{\,'X', 'C', \varepsilon_s, \varepsilon_e\}$ with Dirac character masses $\mu('X') = 1$, $\mu('C') = 2$, $\mu(\varepsilon_s) = \mu(\varepsilon_e) = 0$ and probabilities $\mathbb{P}(C = 'C') = p_\Gamma$ and $\mathbb{P}(C = 'X') = 1 - p_\Gamma$. This alphabet has no terminal characters.

Let *'C'* be the only cleavage character, so $\Gamma = \{\,'C'\}$. Recall that rows refer to masses $m = 0, 1, 2, \ldots$ and columns refer to the states Start, $\bar{\Gamma}, \Gamma$, End, respectively, in matrices Q and $H^{(l)}$. Then, we can derive the 4×4 transition matrix

$$P = \begin{pmatrix} 0 & 1-p_\Gamma & p_\Gamma & 0 \\ 0 & 1-p_\Gamma & p_\Gamma & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

the 3×4 mass emission probability matrix Q with rows for $m = 0, 1, 2$

$$Q = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix},$$

and the 4×4 start distribution matrix

$$\underline{p^0} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Using Lemma 8.5, we compute

$$H^{(0)} = Q \cdot \underline{p^0} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

giving probability 1 to be in start state 0 with emission of mass $m = 0$. The next prefix mass distributions is

$$H^{(1)} = \left(H^{(0)} \cdot P\right) \star Q = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1-p_\Gamma & 0 & 0 \\ 0 & 0 & p_\Gamma & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

since from state 0, we can either transit to a cleavage character state or a non-cleavage character state with emission of mass $m = 2$ or mass $m = 1$, respectively. Note that for both step 0 and step 1, we have probability 0 to be in a final state. From here, we

compute the 2-step prefix mass distributions as

$$H^{(2)} = \left(H^{(1)} \cdot P\right) \star Q = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & (1-p_\Gamma)^2 & (1-p_\Gamma)p_\Gamma & 0 \\ 0 & 0 & 0 & p_\Gamma \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \star \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$
$$= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & (1-p_\Gamma)^2 & 0 & p_\Gamma \\ 0 & 0 & (1-p_\Gamma)p_\Gamma & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

After two steps of the wHMM in Figure 5.2, the probability that the accumulated mass is $m = 0$ or $m = 1$ is zero for all states. To have accumulated mass $m = 2$ (third row in the matrix), the wHMM can either take the steps (state 0→state 1→state 1) corresponding to the string set $\{\varepsilon_s\} \times \bar{\Gamma} \times \bar{\Gamma} \times \{\varepsilon_e\}$ with probability $(1-p_\Gamma)^2$ or the steps (state 0→state 2→state 3) ($\{\varepsilon_s\} \times \bar{\Gamma} \times \{\varepsilon_e\}$) with probability p_Γ.

For mass $m = 3$ the only possibility is the state sequence $(0, 1, 2)$, corresponding to $\{\varepsilon_s\} \times \bar{\Gamma} \times \Gamma \times \{\varepsilon_e\}$. The joint length-mass probability for $l = 1$ is thus the fourth column of the matrix, i.e.

$$f_\circ(1, m) \;=\; \left(H^{(2)}\right)_{m,3} \;=\; (0, 0, p_\Gamma, 0, 0, \cdots)^t,$$

corresponding to the fragment $F_1 =$ *'C'* of mass $\mu(F_1) = 2$.

Example 8.7 (Computation of $\mathbf{f_\circ}$ for standard cleavage schemes). Consider a standard cleavage scheme (Γ, Π) and the corresponding wHMM for F_1 given in Figure 5.4 with state set $E = \{0, \ldots, 4\}$ of size $|E| = 5$ and $T = \{3\}$. If we denote by P_{ij} the one-step transition probability from state i to state j, by $q_j(m)$ the emission probability of mass m in state j, and finally by $h_j^l(m)$ the accumulated mass probability after l steps while being in state j, we get explicit equations derived in the following.

The initial distribution is Dirac, with probability one on the start state 0, so $p_0^0 = 1$ and $p_j^0 = 0$ for $1 \le j \le 4$. We initially have for $l = 0$:

$$h_0^0 \;=\; q_0 \text{ and } h_i^0 \;=\; 0 \quad \text{for any state } i \neq 0.$$

For larger prefix lengths $l \ge 1$, let us have a closer look at the probabilities h_1^l: State 1 can be reached from states $j \in \{0, 1, 4\}$ and its state mass function is given by q_i with $\Sigma_i = \bar{\Gamma}$. The recurrence is then

$$h_1^l \;=\; (P_{01} \cdot h_0^{l-1} \star q_1) \;+\; (P_{11} \cdot h_1^{l-1} \star q_1) \;+\; (P_{41} \cdot h_4^{l-1} \star q_1),$$

and noting that all transition probabilities are $1 - p_\Gamma$, we can simplify this to

$$h_1^l \;=\; (1 - p_\Gamma) \cdot (h_0^{l-1} + h_1^{l-1} + h_4^{l-1}) \star q_1.$$

Similarly, we derive the recurrences for the remaining states:

$$\begin{aligned}
h_0^l &= 0, \\
h_1^l &= (1 - p_\Gamma) \cdot (h_0^{l-1} + h_1^{l-1} + h_4^{l-1}) \star q_1, \\
h_2^l &= p_\Gamma \cdot (h_0^{l-1} + h_1^{l-1} + h_4^{l-1}) \star q_2, \\
h_3^l &= (1 - p_\Pi) \cdot h_2^{l-1} \star q_3, \\
h_4^l &= p_\Pi \cdot h_2^{l-1} \star q_4.
\end{aligned}$$

Finally, the joint length-mass distribution of the first fragment is the fourth column of the matrix $H^{(l+1)}$, thus

$$f_1(l, m) = h_3^{l+1}(m).$$

In the same way, we derive the explicit recurrence equations for the following fragments by using the appropriate quantities of the F_+-wHMM (Figure 5.4, bottom). Then, the initial condition is

$$h_0^0 \;=\; q_0, \text{ and } h_i^0 \;=\; 0 \quad \text{for } i \neq 0,$$

and for step $l \geq 1$,

$$\begin{aligned}
h_0^l &= 0, \\
h_1^l &= (1 - p_\Gamma) \cdot (h_1^{l-1} + h_4^{l-1} + h_5^{l-1}) \star q_1, \\
h_2^l &= p_\Gamma \cdot (1/(1 - p_\Pi) \cdot h_0^{l-1} + h_1^{l-1} + h_4^{l-1} + h_5^{l-1}) \star q_2, \\
h_3^l &= (1 - p_\Pi) \cdot h_2^{l-1} \star q_3, \\
h_4^l &= p_\Pi \cdot h_2^{l-1} \star q_4. \\
h_5^l &= (1 - p_\Pi - p_\Gamma)/(1 - p_\Pi) \cdot h_0^{l-1} \star q_5.
\end{aligned}$$

Finally,

$$f_+(l, m) = h_3^{l+1}(m).$$

Note that we can remove states 0 and 5 from the wHMM (they are used at most once) by specifying a more complicated initial condition after step 2 in this case. However, the wHMM construction is general and can be applied to more complicated string and cleavage models.

8.2. Recurrence Equations

While the wHMM framework is easily generalizable, we can also give a more combinatorial derivation; the resulting recurrence equations can later be used to derive more space- and time-efficient algorithms.

As for the length distributions, our considerations are based on the structural Lemma 5.16; the resulting recurrence equations are basically the same as their counterparts for the length distributions given in Theorem 6.4.

To ease the exposition and avoid explicit writing of the convolution, we introduce the following notation.

Definition 8.8 (Convolution of length-mass distributions). Let (Σ, μ) be a weighted alphabet and let $\Theta \subseteq \Sigma$ be a sub-alphabet. Let

$$r(\Theta, m) \;:=\; \mathbb{P}(C \in \Theta, \mu(C) = m) \;=\; \mathbb{P}(C \in \Theta) \cdot \mathbb{P}(\mu(C) = m \mid C \in \Theta)$$

be the probability that a random character chosen from Σ is an element of Θ and has mass m.

The convolution of $r(\Theta, \cdot)$ with a joint fragment length-mass distribution $f(l-1, \cdot)$ is then defined as

$$\left(f_\circ(l-1,\cdot) \star r(\Theta,\cdot)\right)(m) := \sum_{\sigma \in \Theta} \sum_{m' \in \mathbb{N}_0} f_\circ(l-1, m-m') \cdot \mathbb{P}(\mu(\sigma) = m') \cdot \mathbb{P}(C = \sigma).$$

Note that the explicit summation over the character masses is in fact finite due to the finite character mass range and is bounded by $\mu_{\max} - \mu_{\min} + 1$ summands. A convolution with a character mass function can thus be performed in time $\mathcal{O}(\mu_{\max} - \mu_{\min})$.

Using the structural Lemma 5.16 and Theorem 6.4, the length-mass distributions can be computed using recurrence equations of the same structure as for the length distributions. Similarly, let us denote by $g(\cdot,\cdot)$ the (defected) joint length-mass distribution for weighted strings in $\mathcal{G}$, starting with a prohibition character.

Theorem 8.9 (Recurrence for $f_\circ$). *Consider $f_\circ(l,m) = \mathbb{P}(L_\circ = l, \mu(F_\circ) = m)$, the joint length-mass distribution of a cleavage fragment $F_\circ$, and let $g(l,m)$ be the probability that a fragment suffix, starting with a prohibition character, has length l and mass m.*

The length-mass distribution of the first fragment is given by

$$f_1(l,m) = \begin{cases} (1-p_\Pi) \cdot r(\Gamma, m) & \text{if } l = 1, \\ \left(f_1(l-1,\cdot) \star r(\bar{\Gamma},\cdot)\right)(m) & \\ \quad + \left(g(l-1,\cdot) \star r(\Gamma,\cdot)\right)(m) & \text{if } l > 1. \end{cases}$$

The distribution of following fragments is

$$f_+(l,m) = \begin{cases} r(\Gamma \cap \bar{\Pi}, m) & \text{if } l = 1, \\ \frac{1}{\bar{\pi}} \cdot \left(f_1(l-1,\cdot) \star r(\bar{\Gamma} \cap \bar{\Pi},\cdot)\right)(m) & \\ \quad + \frac{1}{\bar{\pi}} \cdot \left(g(l-1,\cdot) \star r(\Gamma \cap \bar{\Pi},\cdot)\right)(m) & \text{if } l > 1. \end{cases}$$

The distribution of fragment suffixes starting with a prohibition character is

$$g(l,m) = \begin{cases} (1-p_\Pi) \cdot r(\Gamma \cap \Pi, m) & \text{if } l = 1, \\ \left(f_1(l-1,\cdot) \star r(\bar{\Gamma} \cap \Pi,\cdot)\right)(m) & \\ \quad + \left(g(l-1,\cdot) \star r(\Gamma \cap \Pi,\cdot)\right)(m) & \text{if } l > 1. \end{cases}$$

The previous theorem is only valid for non-terminal alphabets: We completely ignore the potential presence of terminal characters. As mentioned in Section 5.2, these characters do not contribute to the length of a fragment; they nevertheless contribute to its mass. Considering this additional mass is straightforward: We convolve the corresponding mass distributions $\mathcal{L}(\mu(\varepsilon_s))$ and $\mathcal{L}(\mu(\varepsilon_e))$ to $f_\circ(l,\cdot)$ for each fragment length l. For simplifying the exposition, we will ignore terminal characters for the remainder of this chapter.

8.3. Finite Strings

For the case of finite string length, we follow our previous conventions and use the following notation (cf. Section 6):

$$f_1^{\langle\ell\rangle}(l,m) := \mathbb{P}\left(L_1^{\langle\ell\rangle} = l, \mu\big(F_1^{\langle\ell\rangle}\big) = m\right),$$
$$f_+^{\langle\ell\rangle}(l,m) := \mathbb{P}\left(L_i^{\langle\ell+k\rangle} = l, \mu\big(F_i^{\langle\ell+k\rangle}\big) = m \mid C_{i-1} = k\right) \text{ for any } i \geq 2 \text{ and any } k \in \mathbb{N}.$$

As for the length distributions in finite strings, the second definition is in fact independent of $i \geq 2$ and k, and defines the conditional joint distribution of (L_+, μ_{F_+}) given that there are ℓ characters left in the string.

Lemma 8.10 (Computation of $\mathbf{f}^{\langle\ell\rangle}$ via wHMMs). *For $l < \ell$, we have $f_\circ^{\langle\ell\rangle}(l,m) = f_\circ(l,m)$ for all masses m. For the last fragment and string length ℓ, we get*

$$f_\circ^{\langle\ell\rangle}(\ell,m) \;=\; \sum_{i \notin T} h_i^\ell(m)$$

for T and $h_i^\ell(m)$ of the appropriate wHMM for any cleavage scheme.

Proof. For $l < \ell$, there is no difference to the semi-infinite string. The fragment ends after position ℓ irrespective of the current state; therefore, summing $h_i^\ell(m)$ over all non-final states i leads to the desired marginal. □

Using the structural Lemma 5.16 for sets of feasible fragment strings, we can also derive recurrence equations for computing the finite length-mass distributions. These are almost the same as those for the infinite case, except for different initial conditions.

Lemma 8.11 (Computation of $\mathbf{f}^{\langle\ell\rangle}$ via recurrences). *With the notation of Definition 8.8, the length-mass distribution of a first fragment for finite string length ℓ is given by the recurrence equations*

$$f_1^{\langle\ell\rangle}(l,m) = \begin{cases} f_1(l,m) & \text{if } l < \ell, \\ r(\Sigma,m) & \text{if } l = \ell = 1, \\ \left(f_1^{\langle\ell-1\rangle}(\ell-1,\cdot) \star r(\bar{\Gamma},\cdot)\right)(m) & \\ \quad + \left(g^{\langle\ell-1\rangle}(\ell-1,\cdot) \star r(\Gamma,\cdot)\right)(m) & \text{if } l = \ell > 1. \end{cases}$$

The distribution of following fragments is

$$f_+^{\langle\ell\rangle}(l,m) = \begin{cases} f_+(l,m) & \text{if } l < \ell, \\ r(\bar{\Pi},m) & \text{if } l = \ell = 1, \\ \frac{1}{\pi}\cdot\left(f_1^{\langle\ell-1\rangle}(\ell-1,\cdot)\star r(\bar{\Gamma}\cap\bar{\Pi},\cdot)\right)(m) & \\ +\,\frac{1}{\pi}\cdot\left(g^{\langle\ell-1\rangle}(\ell-1,\cdot)\star r(\Gamma\cap\bar{\Pi},\cdot)\right)(m) & \text{if } l = \ell > 1. \end{cases}$$

The distribution of fragment suffixes starting with a prohibition character is

$$g^{\langle\ell\rangle}(l,m) = \begin{cases} g(l,m) & \text{if } l < \ell, \\ r(\Pi,m) & \text{if } l = \ell = 1, \\ \left(f_1^{\langle\ell-1\rangle}(\ell-1,\cdot)\star r(\bar{\Gamma}\cap\Pi,\cdot)\right)(m) & \\ +\,\left(g^{\langle\ell-1\rangle}(\ell-1,\cdot)\star r(\Gamma\cap\Pi,\cdot)\right)(m) & \text{if } l = \ell > 1. \end{cases}$$

8.4. Related Distributions

Several probabilities and distributions derived from the length-mass distributions turn out to be helpful later. In particular, we briefly investigate the two marginal distributions, i.e. the length and the mass distributions of fragments, the mass avoidance probabilities of a fragment of certain length l not taking mass m, and conditional probabilities that give the distributions of fragment length, given the mass, and vice-versa.

Length distributions as marginals. From the joint distribution of fragment length and mass, we can derive two other distributions by taking the marginals: By summing over all masses m, we re-derive the distribution of fragment lengths:

$$u_\circ(l) = \sum_{m\in\mathbb{N}_0} f_\circ(l,m),$$

and similarly for finite string length. Using the $f_\circ(\cdot,\cdot)$-recurrences of Theorem 8.9, we also re-derive the corresponding recurrence equations of Theorem 6.4 for $u_\circ(\cdot)$.

Mass distributions as marginals. In the same manner, we can derive the distribution of fragment masses regardless of their length by taking the other marginal. Let us denote these marginal distributions by re-using the symbol f in a straightforward way:

$$f_1(m) := \mathbb{P}(\mu_{F_1} = m),$$
$$f_+(m) := \mathbb{P}(\mu_{F_+} = m).$$

Obviously, we can simply sum over all possible fragment lengths at mass m to get the mass distribution:

Proposition 8.12. *(Fragment mass distribution) The distributions of fragment masses for first and following fragments under any cleavage scheme are given by*

$$f_\circ(m) \;=\; \sum_{l\in\mathbb{N}} f_\circ(l,m).$$

Using the recurrences for the length-mass distributions, we can also establish similar recurrences for the mass distributions.

Lemma 8.13 (Recurrence for $\mathbf{f_\circ}$). *Similar to Definition 8.8, define the convolution of a fragment mass distribution with $r(\Theta,\cdot)$ by*

$$\left(f_\circ(\cdot)\star r(\Theta,\cdot)\right)(m) \;:=\; \sum_{\sigma\in\Theta}\sum_{m'\in\mathbb{N}_0} f_\circ(m-m')\cdot\mathbb{P}(C=\sigma)\cdot\mathbb{P}(\mu(\sigma)=m').$$

The mass distributions $f_\circ(\cdot)$ for any cleavage scheme can be computed by the recurrence equation

$$\begin{aligned}
f_1(m) &= \sum_{m'\le m} \left(f_1(m-m')\star r(\bar\Gamma,m') \;+\; g(m-m')\star r(\Gamma,m')\right)\\
f_+(m) &= \sum_{m'\le m} \left(f_1(m-m')\star r(\bar\Gamma\cap\bar\Pi,m') \;+\; g(m-m')\star r(\Gamma\cap\bar\Pi,m')\right)\\
g(m) &= \sum_{m'\le m} \left(f_1(m-m')\star r(\bar\Gamma\cap\Pi,m') \;+\; g(m-m')\star r(\Gamma\cap\Pi,m')\right)
\end{aligned}$$

with $f_\circ(m)=0$ and $g(m)=0$ for $m<\mu_{\min}$.

Proof. We first observe that all recurrences of length-mass distributions are linear and have constant coefficients, thus, the structure of these recurrence translates to partial sums of the involved quantities. Using the recurrences of Theorem 8.9, summing over all possible fragment lengths l and noting that $f_\circ(l,m)=g(l,m)=0$ if $m<0$ immediately yields the stated results. □

This last recurrence equations enables us to compute the mass distribution without the length-mass distribution.

Mass avoidance probabilities. In subsequent chapters, we will need the probability that a fragment has length l and *not* mass m.

Definition 8.14 (Mass avoidance probabilities). The *mass avoidance* probabilities of a fragment having length l and not mass m are defined as

$$\begin{aligned}
\bar f_1(l,m) &:= \mathbb{P}(L_1=l,\mu(F_1)\neq m),\\
\bar f_+(l,m) &:= \mathbb{P}(L_+=l,\mu(F_+)\neq m),
\end{aligned}$$

and similarly as $\bar f_\circ^{\langle\ell\rangle}$ for fragments whose length is bounded by ℓ.

The avoidance probabilities can be computed from the length and the length-mass distributions of fragments.

Lemma 8.15 (Mass avoidance probabilities). *The mass avoidance probabilities for any cleavage scheme are given by*

$$\bar{f}_\circ(l,m) \;=\; u_\circ(l) - f_\circ(l,m),$$

and similarly for $\bar{f}_\circ^{\langle\ell\rangle}$.

Proof. We have

$$\begin{aligned}\bar{f}_\circ(l,m) &= \sum_{m'\neq m} f_\circ(l,m') \\ &= \left(\sum_{m'\in\mathbb{N}_0} f_\circ(l,m')\right) - f_\circ(l,m) \\ &= u_\circ(l) - f_\circ(l,m),\end{aligned}$$

and similarly for the finite string case. □

This lemma can be informally stated in a combinatorial fashion as: "The fragments of length l and not mass m are all fragments of length l minus the ones of the same length having mass m."

Conditional distributions. As a last result, we investigate the distributions of the lengths of fragments with known mass and the mass distribution of fragments with known length. These distributions are easily derived from the joint distributions and their marginals.

Lemma 8.16 (Conditional distributions). *Using the joint length-mass distributions* $f_\circ(\cdot,\cdot)$ *and their two marginals* $u_\circ(\cdot)$ *and* $f_\circ(\cdot)$*, the conditional distributions of fragment length, given the masses, and of mass, given the lengths, are*

$$\begin{aligned}\mathbb{P}\left(L_\circ = l \mid \mu(F_\circ) = m\right) &= \frac{\mathbb{P}\left(L_\circ = l,\, \mu\left(F_\circ\right) = m\right)}{\mathbb{P}\left(\mu\left(F_\circ\right) = m\right)} \;=\; \frac{f_\circ(l,m)}{f_\circ(m)}, \text{ and} \\ \mathbb{P}\left(\mu(F_\circ) \mid L_\circ = l\right) &= \frac{\mathbb{P}\left(L_\circ = l,\, \mu\left(F_\circ\right) = m\right)}{\mathbb{P}\left(L_\circ = l\right)} \;=\; \frac{f_\circ(l,m)}{u_\circ(l)}.\end{aligned}$$

These equations hold for any cleavage scheme.

Proof. By using the identity $\mathbb{P}(X \in A,\, Y \in B) = \mathbb{P}(X \in A \mid Y \in B) \cdot \mathbb{P}(Y \in B)$ for random variables X, Y and appropriate sets A, B. □

Example 8.17 (Conditional probabilities for cleavage characters). Using Swiss-Prot frequencies and tryptic digestion, we compute the conditional length distribution, given that the fragment has mass $\mu('K')$:

$$\mathbb{P}(L_1 = l \mid m = \mu('K')) \;=\; \begin{cases} 1, & \text{for } l = 1, \\ 0, & \text{else.} \end{cases}$$

Further, the probability of fragment mass m, given the fragment has length 1 for the first fragment is

$$\mathbb{P}(\mu(F_1) = m \mid L_1 = 1) \;=\; \begin{cases} 0.526\ldots, & \text{for } m = \mu('K'), \\ 0.474\ldots, & \text{for } m = \mu('R'), \\ 0, & \text{else.} \end{cases}$$

Not surprising, a fragment of length 1 can only be a cleavage character.

8.5. Implementation

So far, we were not concerned with the problem of computing the distributions and access their values on a computer in reasonable time. Unlike the fragment length distributions, we can neither approximate the length-mass distributions by a known function, nor give closed formulas for them. In order to access each value of these distributions in constant time, we thus need to compute all values and store them in an appropriate data structure. Of course, we are not able to store every combination of values for every possible length and mass. This is not a problem in practice: The maximal protein length $\ell_{\max}$ gives an upper bound for the possible fragment length; it also gives an upper bound for the possible fragment mass.

In the context of mass spectrometry, we know the maximal molecular mass that can be detected by the MS instrument; it is usually much smaller than the maximal protein mass. If we denote the maximal detectable mass by $m_{\max}$, we can easily estimate an upper bound $l_{\max}$ for the maximal length of fragments detectable by the instrument:

$$l_{\max} \;=\; \left\lceil \frac{m_{\max}}{\mu_{\min}} \right\rceil .$$

Terminal masses $\mu(\varepsilon_s), \mu(\varepsilon_e)$ may be subtracted from $m_{\max}$ if we use terminal-extended alphabets. The upper bound is valid for all fragment masses up to $m_{\max}$ and thus gives an upper bound for the number of summands in the computation of $f_\circ(\cdot,\cdot)$. Of course, we can compute upper bounds for the number of summands for each mass. We will not investigate this here, because $l_{\max}$ is quite small in practice (10–100, see below) and we do not get a considerable improvement in computation time or space requirements. Note that the upper bound is independent of the mass precision Δ_m.

Time complexity of length-mass probabilities. Each entry of $f_\circ(\cdot,\cdot)$ and $g(\cdot,\cdot)$ is computed in $\mathcal{O}(\mu_{\max}-\mu_{\min})$ time using the recurrence equations of Theorem 8.9 and standard dynamic programming techniques. It involves summation of at most $2(\mu_{\max}-\mu_{\min})$ non-zero terms for the convolution of masses. Storing these entries in an $l_{\max}\times m_{\max}$ table, computing an entry at row l and column m involves a look-back of at most one row and $\mu_{\max}$ columns.

All tables can thus be computed in time

$$\mathcal{O}(l_{\max}\cdot m_{\max}\cdot(\mu_{\max}-\mu_{\min})),$$

for maximal fragment length $l_{\max}$ and maximal mass $m_{\max}$. Note that the stated time complexities also depend on the mass precision: Masses are always given as scaled and rounded integers, so increasing the precision by one decimal, i.e. from Δ_m to $\Delta_m/10$, changes both $m_{\max}$ and $\mu_{\max}-\mu_{\min}$ by a factor of 10. More precisely, the previous time complexity thus reads

$$\mathcal{O}\left(l_{\max}\cdot\frac{m^*_{\max}}{\Delta_m}\cdot\frac{{\mu_{\max}}^*-{\mu_{\min}}^*}{\Delta_m}\right),$$

if $m^*_{\max}$ and ${\mu_{\max}}^*, {\mu_{\min}}^*$ refer to the masses in Da, i.e., without scaling. We will stick to the scaled masses for fixed precision and just keep in mind that changing the mass precision also changes the time complexity.

Further, we assume the alphabet Σ to be finite and small. As a consequence, we do not bother about the sizes of the character sets Γ and Π in the time complexity analysis, although these sizes obviously influence the computation time. More precisely, when looking at both Theorem 8.9 and Definition 8.8, we see that computing one entry of $f_\circ(\cdot,\cdot)$, given the previous ones, not only takes time $\mathcal{O}(\mu_{\max}-\mu_{\min})$ but also involves summation over the character subsets of Σ, so the number of summands is bounded by the alphabet size $|\Sigma|$. Considering this, the full time complexity for computing the two length-mass distributions is

$$\mathcal{O}\left(l_{\max}\cdot\frac{m^*_{\max}}{\Delta_m}\cdot\frac{{\mu_{\max}}^*-{\mu_{\min}}^*}{\Delta_m}\cdot|\Sigma|\right).$$

Since the alphabet is small for all biological applications (20 for proteins, 4 for DNA/RNA), we will also neglect this term in all further analyses.

Finally, adjustments for finite string lengths can be computed in time $\mathcal{O}(\mu_{\max}-\mu_{\min})$ for each entry, given the $f_\circ(\cdot,\cdot)$ tables.

Memory requirements of length-mass probabilities. The length-mass distributions are used in two different scenarios: Either, we want to access these distributions for some statistical computations and need access to each entry in constant time; we then have to keep all entries in memory. In subsequent chapters, however, we only need these distributions for one certain mass at a time, increasing the mass by one in each step. Then, we only need to keep those parts of the distributions in memory that give the entries for this particular mass m and allow computation of the entries of mass $(m+1)$.

Note again that m is a scaled mass and its actual value depends on the original mass m^* in Da to be considered and the mass precision Δ_m.

For the first scenario, we obviously need $\mathcal{O}(l_{\max} \cdot m_{\max})$ memory to store each $f_\circ(\cdot,\cdot)$ table. Let us assume Swiss-Prot frequencies and average masses with a precision of 0.1 Da, see Table 1.1 for the values. Let us further assume that the MS instrument has a maximal detectable mass of 3 500 Da (a usual value for MALDI-TOF-MS). We compute $m_{\max} = 35\,000$ columns and $l_{\max} = \lceil 35\,000/571 \rceil = 62$ rows for each distribution. Using doubles of 8 bytes each, the two tables require $2 \cdot 62 \cdot 35\,000 \cdot 8$ bytes. This is about 33.1 MB. During computation, we also need the $g(\cdot,\cdot)$ table of about 17 MB.

For the second scenario, where we only access one column at a time, we need the $\mu_{\max}$ previous columns of both $f_\circ(\cdot,\cdot)$ and $g(\cdot,\cdot)$ to compute $f_\circ(\cdot,\cdot)$ using the recurrences of Theorem 8.9. With the assumptions of the previous paragraph and $\mu_{\max} = 1\,860$ (tryptophan with $\Delta_m = 0.1$), this corresponds to $2 \cdot 62 \cdot (2\,\mu_{\max}) \cdot 8$ bytes, about 3.5 MB for both tables.

Mass avoidance probabilities. Given the implementation for the length distributions of Section 6.6, we can compute the mass avoidance probabilities in the same time- and space complexity as the length-mass distributions. Note that for l outside the length range implied by the maximal fragment mass, the avoidance probability is equal to the length probability:

$$\bar{f}_\circ(l,m) \;=\; u_\circ(l) \text{ if } l > l_{\max} \text{ and } m \le m_{\max}.$$

Mass distributions. The mass distributions $f_\circ(\cdot)$ can be computed in time $\mathcal{O}(\mu_{\max} - \mu_{\min})$ for each entry using the recurrence in Lemma 8.13 and dynamic programming, or in time $\mathcal{O}(l_{\max})$ for each entry, summing over the lengths in the length-mass distributions $f_\circ(\cdot,\cdot)$.

We need at most $m_{\max}$ entries of each of these distributions, so the space requirement is $\mathcal{O}(m_{\max})$. Using the previous assumptions, this means $2 \cdot 35\,000 \cdot 8$ bytes, about 547 kB for both $f_\circ(\cdot)$ tables.

8.6. Evaluation on Swiss-Prot

The probability $f_\circ(l,m)$ for a given mass m and length l depends on the number of combinations of l amino acids with masses summing up to exactly m. The problem of finding the number of decompositions $d(m)$ of mass m has been studied in [25, 26].

For given fragment length l, the mass distribution $f_\circ(l,\cdot)$ is a convolution of l character mass distributions. Due to the character composition constraints for fragments, these individual distributions are not independent. Nevertheless, the dependence is only weak and we would assume that a weaker version of the Central Limit Theorem applies and the mass distribution approaches a Gaussian distribution for large l.

Using our standard setting TryptSwissProt– Swiss-Prot database amino acid frequencies, the standard cleavage scheme of Trypsin and a mass precision of $\Delta_m = 0.1$, we

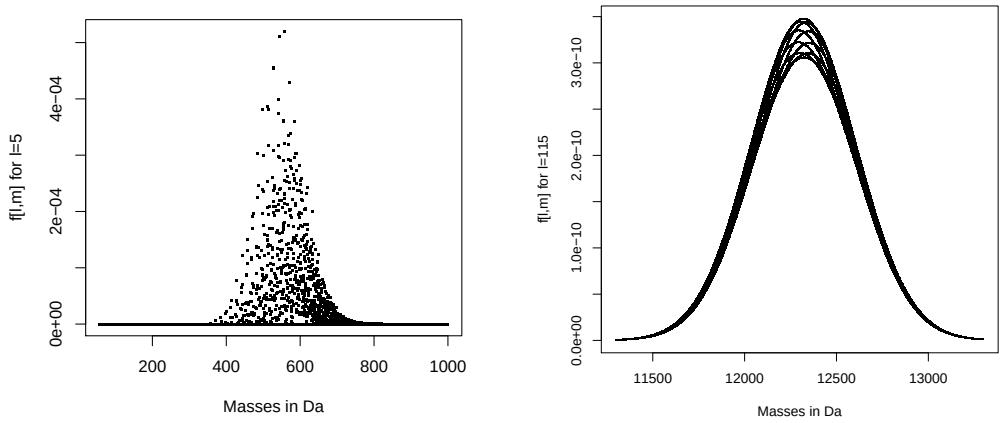

Figure 8.1.: $f_1(l, \cdot)$ for $l = 5$ (left) and $l = 115$ (right), mass precision $\Delta_m = 0.1$.

obtain the first fragment's joint length-mass distribution $f_1(l, \cdot)$ for length $l = 5$ (Figure 8.1, left), length $l = 25$ (Figure 8.2, left), and length $l = 115$ (Figure 8.1, right). As expected, the mass distribution indeed visually approaches a Gaussian for increasing fragment lengths. Nevertheless, fragments of greater length are extremely rare (compare the length distributions in Figure 6.2, left and right). For the frequent fragment lengths below 30, the combinatorics of the mass composition is still predominant and the distributions cannot be approximated by a continuous function (such as a Gaussian density function).

In [25], it was also observed that the number $d(m)$ of decompositions of mass m is periodic in m. This behavior is at least partly explained by a further study of the related money-changing problem, see for example Example 1 on page 99 of [130]. Decomposition of an integer was also studied in [14, 15]. Not surprising, periodicity is also observed in the length-mass distributions: For each length l, the function $f_\circ(l, m)$ is periodic in m with period 20. This effect is shown in Figure 8.2 (right): The plot shows the function $f_1(25, m)$ for masses having remainder $r = 0, 1, 2$ when divided by 20, so $r = m \mod 20$. The period 20 suggests to be related to the size of the amino acid alphabet, which is also 20. However, decreasing the alphabet size to 19, 18, or 17 by removing non-cleavage-non-prohibition characters $\sigma \in \bar{\Gamma} \cap \bar{\Pi}$ from the alphabet did not change the period of the corresponding function.

For following fragments of length $l = 15$, we computed expected counts $c_+(15, m)$ for each mass m from the length-mass distribution by multiplying by the number A_{15} of fragments of length $l = 15$ and dividing by the length probability $u_+(15)$:

$$c_+(15, m) \;:=\; f_+(15, m) \cdot \frac{A_{15}}{u_+(15)}.$$

These counts are plotted together with the empirical counts in Figure 8.3 (left). There are 141 842 fragments of length 15 in the Swiss-Prot database using tryptic digestion;

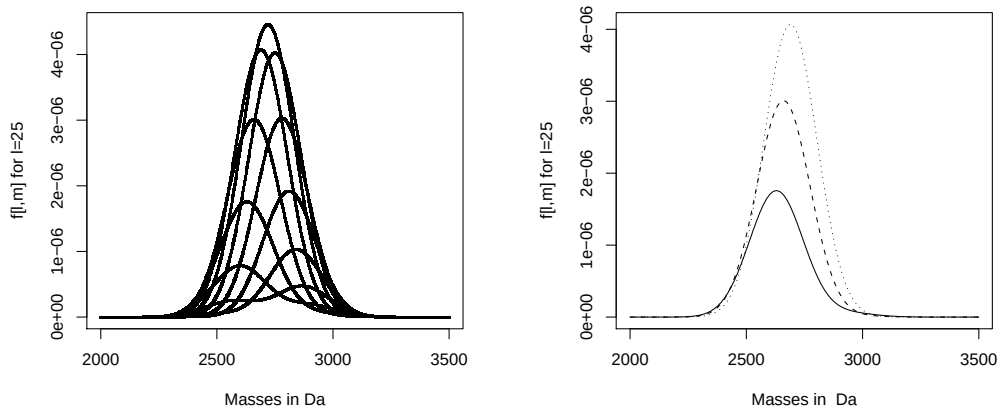

Figure 8.2.: Left: $f_1(l, \cdot)$ for $l = 25$. Right: $f_1(l, \cdot)$ for $l = 25$ masses taken modulo 20 and three remainders $r = m \mod 20$. Solid: $r = 0$, dashed: $r = 1$, dotted: $r = 2$. Both mass precisions $\Delta_m = 0.1$.

their masses range from 955.6 Da to 2 114.8 Da. The maximal number of fragments of a certain mass was slightly under 200. This number is too small to cover the combinatorial effects in the estimation; from this estimation, it is not possible to deduce the quality of the model's fit to the data.

The fragment mass distribution $f_1(m)$ is given in Figure 8.3 (right) for masses ranging from 1 000 Da to 1 500 Da, also showing the periodicity of this function. The smaller figure gives $f_1(m)$ in the range from 120 Da to 500 Da. For better readability, only probabilities larger than 10^{-4} are plotted. The two points in the upper left corner correspond to the two tryptic single-character fragments 'K' and 'R'.

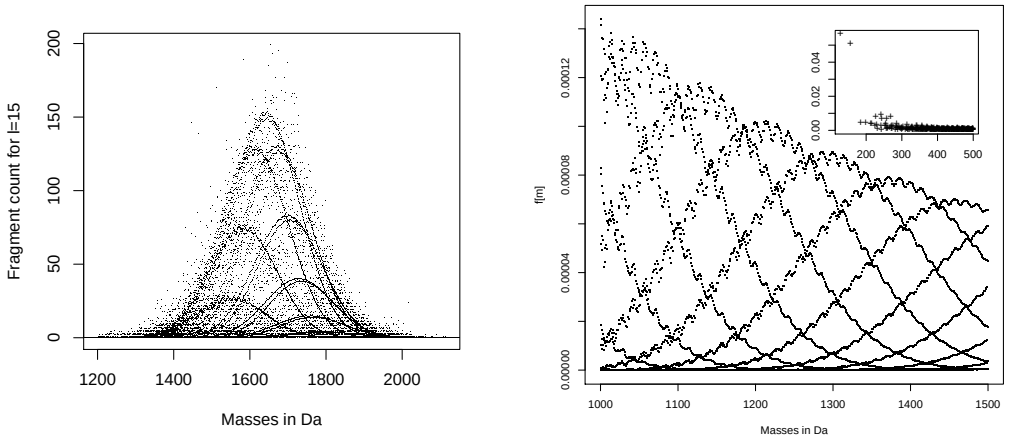

Figure 8.3.: Left: Fragment counts for $l = 15$ and empirical Swiss-Prot fragment counts (see text for details). Right: Mass distribution $f_1(m)$ in two mass ranges. Both mass precisions $\Delta_m = 0.1$.

9. Mass Occurrence Probabilities

One of the main goals of this first part of the thesis is to identify and compute statistics of mass fingerprints relevant for computing the significance of a protein identification by peptide mass fingerprinting. As a last important statistic, we investigate the probability that a certain mass m occurs in the fragmentation of a finite random weighted string. These probabilities are used in the second part of the thesis to compute the contribution of a single peak to the overall score of a protein identification. The following considerations are valid for any cleavage scheme.

Let us first translate this *mass occurrence problem* into a stochastically more suitable context: Instead of looking at the fragmentation and the fragments' masses directly, we introduce the waiting time $T(m) \equiv T(S, m)$ for the first occurrence of a fragment of mass m in an infinite random weighted string (S, μ): It is the first index in S where a fragment $F_\circ$ of mass $\mu(F_\circ) = m$ ends:

$$T(m) \;:=\; \min\{j \in \mathbb{N} \,|\, \exists k \in \mathbb{N} : C_k = j,\, \mu(F_k) = m\},$$

where we define $T(m) = \infty$ if the minimum is taken over the empty set, i.e., if mass m is not decomposable as a fragment.

The occurrence probability of mass m can then be written in terms of $T(m)$: The random weighted string (S, μ) contains a fragment of mass m in its prefix up to letter ℓ if and only if the waiting time for this fragment is less or equal ℓ. Note that the last fragment starting before ℓ is not taken into consideration unless it ends exactly in ℓ.

Of course, proteins do not have infinite length and we need to adapt the waiting time to the case of finite protein length. In this case, we have a last fragment with different structure in the fragmentation. Let $T^{\langle\ell\rangle}(m)$ denote the waiting time adapted to finite string length ℓ, i.e. the suffix from $C_{N^{\langle\ell\rangle}-1} + 1$ to ℓ is considered as a fragment:

$$T^{\langle\ell\rangle}(m) \;:=\; \min\left\{ j \in \mathbb{N} \;\middle|\; j \le \ell \text{ and } \exists k \in \mathbb{N} : C_k^{\langle\ell\rangle} = j,\, \mu\left(F_k^{\langle\ell\rangle}\right) = m \right\},$$

where again $T^{\langle\ell\rangle}(m) = \infty$ if the minimum is taken over the empty set.

Definition 9.1 (Mass occurrence probability). The *occurrence probability* $p^{\langle\ell\rangle}(m)$ of mass m in the fragmentation of a finite random weighted string of length ℓ under any cleavage scheme is defined as

$$p^{\langle\ell\rangle}(m) \;:=\; \mathbb{P}\left(T^{\langle\ell\rangle}(m) \le \ell\right),$$

the probability that a fragment of mass m occurs before the string ends.

Remark. Note that $p^{\langle\ell\rangle}(m)$ is *not* the cumulative distribution function of the waiting time $T(m)$ or $T^{\langle\ell\rangle}(m)$. Each such occurrence probability is defined on a different probability space (one for each finite string length ℓ). In particular, the occurrence probability will not necessarily increase monotonically from 0 to 1. The cumulative distribution functions are given by the probabilities $\mathbb{P}(T(m) \leq l)$ and $\mathbb{P}(T^{\langle\ell\rangle}(m) \leq l)$, respectively.

Note also that the random variable $T(m)$ and its finite counterpart may be defective, i.e. their distribution functions may not increase to 1 but stay constant. We will investigate these defective cases in more detail in Section 9.2.

In Section 9.5, we will investigate the case that not the protein length but its mass, the so-called *parent mass*, is given.

Let us first investigate the case of occurrence probabilities in a protein of given length. Here, we need both the length distributions and the length-mass distributions of fragments. They can be computed either within the wHMM framework as described in Sections 6.1 and 8.1 or by the recurrence equations given in Sections 6.2 and 8.2.

9.1. Recurrence Equations

For computing $p^{\langle\ell\rangle}(m)$, we make use of the non-occurrence probability $\bar{p}^{\langle\ell\rangle}(m)$ of the complementary event that no fragment of mass m occurs in the fragmentation of $S^{\langle\ell\rangle}$:

$$\bar{p}^{\langle\ell\rangle}(m) \;:=\; \mathbb{P}\left(T^{\langle\ell\rangle}(m) > \ell\right) \;=\; \mathbb{P}\left(\mu\left(F_1^{\langle\ell\rangle}\right) \neq m, \ldots, \mu\left(F_{N^{\langle\ell\rangle}}^{\langle\ell\rangle}\right) \neq m\right),$$

from which we recover the mass occurrence probability by

$$p^{\langle\ell\rangle}(m) \;=\; 1 - \bar{p}^{\langle\ell\rangle}(m).$$

The main idea for computing the non-occurrence probabilities is the following: For $T^{\langle\ell\rangle}(m)$ to be greater than ℓ, the first fragment of $(S^{\langle\ell\rangle}, \mu)$ must not have mass m and the remaining suffix of length $\ell - L_1^{\langle\ell\rangle}$ must not contain a fragment of mass m. Given the first fragment's length $L_1^{\langle\ell\rangle}$, its mass becomes independent of all the following fragments' masses. We have already seen that the various fragment distributions differ for first and following fragments. The probability for the first fragment to have mass m is usually different from the corresponding probability of a following fragment to have mass m. Therefore, to apply the above argument again on the remaining suffix $S^{\langle\ell\rangle}_{L_1^{\langle\ell\rangle}+1:\ell}$ of $S^{\langle\ell\rangle}$ of length $\ell' = \ell - L_1^{\langle\ell\rangle}$, let $T_+^{\langle\ell'\rangle}(m)$ denote the waiting time for a fragment of mass m in this ℓ'-suffix of $S^{\langle\ell\rangle}$, and let

$$\bar{p}_+^{\langle\ell'\rangle}(m) \;:=\; \mathbb{P}\left(T_+^{\langle\ell'\rangle}(m) > \ell'\right)$$

be the corresponding non-occurrence probability.

For computing $\bar{p}_+^{\langle\ell'\rangle}(m)$, an argument similar to the one given above applies: The probability not to have a fragment of mass m in this suffix is the probability that the first fragment in this suffix does not have mass m and the remaining suffix of length $\ell' - L_2^{\langle\ell\rangle}$ does not contain such a fragment.

Full-history recurrences. Formalizing the stated idea immediately gives a recurrence equation for the occurrence probabilities for given protein length.

Theorem 9.2 (Mass occurrence probabilities for given protein length). *Let $(S^{\langle\ell\rangle}, \mu)$ be a finite random weighted string of length ℓ cleaved with any cleavage scheme (Γ, Π). The probability that $S^{\langle\ell\rangle}$ does not have a fragment of mass m is a convolution over string length:*

$$\bar{p}^{\langle\ell\rangle}(m) \;=\; \sum_{l=1}^{\ell} \bar{p}_{+}^{\langle\ell-l\rangle}(m) \cdot \bar{f}_{1}^{\langle\ell\rangle}(l, m).$$

Similarly, the non-occurrence probability in an ℓ-suffix can be computed by a convolution over the suffix-length, using the appropriate length-mass distribution for following fragments:

$$\bar{p}_{+}^{\langle\ell\rangle}(m) \;=\; \sum_{l=1}^{\ell} \bar{p}_{+}^{\langle\ell-l\rangle}(m) \cdot \bar{f}_{+}^{\langle\ell\rangle}(l, m).$$

The initial conditions are $\bar{p}^{\langle\ell\rangle}(m) = 1$ for $\ell = 0$ and $\bar{p}_{+}^{\langle\ell\rangle}(m) = 1$ for $\ell = 0$.

Finally, taking the probabilities

$$p^{\langle\ell\rangle}(m) = 1 - \bar{p}^{\langle\ell\rangle}(m), \text{ and}$$
$$p_{+}^{\langle\ell\rangle}(m) = 1 - \bar{p}_{+}^{\langle\ell\rangle}(m),$$

gives the mass occurrence probabilities of mass m in a string of length ℓ and in an ℓ-suffix, respectively.

Proof. The main observation for the proof is that although the fragment masses are not independent, as we deal with finite string length, the mass of a fragment becomes independent of the remaining fragments' masses once its length is known.

$$\begin{aligned}
\mathbb{P}\left(T^{\langle\ell\rangle}(m) > \ell\right) &= \mathbb{P}\left(\mu\left(F_1^{\langle\ell\rangle}\right) \neq m, \ldots, \mu\left(F_{N^{\langle\ell\rangle}}^{\langle\ell\rangle}\right) \neq m\right) \\
&= \sum_{l=1}^{\ell} \mathbb{P}\left(\mu\left(F_1^{\langle\ell\rangle}\right) \neq m, \ldots, \mu\left(F_{N^{\langle\ell\rangle}}^{\langle\ell\rangle}\right) \neq m, L_1^{\langle\ell\rangle} = l\right) \\
&= \sum_{l=1}^{\ell} \mathbb{P}\left(\mu\left(F_1^{\langle\ell\rangle}\right) \neq m, \ldots, \mu\left(F_{N^{\langle\ell\rangle}}^{\langle\ell\rangle}\right) \neq m \mid L_1^{\langle\ell\rangle} = l\right) \cdot \mathbb{P}\left(L_1^{\langle\ell\rangle} = l\right).
\end{aligned}$$

We can now use the conditional independence to get

$$\begin{aligned}
\mathbb{P}\left(T^{\langle\ell\rangle}(m) > \ell\right) \;&=\; \sum_{l=1}^{\ell} \mathbb{P}\left(\mu\left(F_2^{\langle\ell\rangle}\right) \neq m, \ldots, \mu\left(F_{N^{\langle\ell\rangle}}^{\langle\ell\rangle}\right) \neq m \mid L_1^{\langle\ell\rangle} = l\right) \\
&\qquad \cdot \mathbb{P}\left(L_1^{\langle\ell\rangle} = l, \mu\left(F_1^{\langle\ell\rangle} \neq m\right)\right) \\
&= \sum_{l=1}^{\ell} \bar{p}_{+}^{\langle\ell-l\rangle}(m) \cdot \bar{f}_{1}^{\langle\ell\rangle}(l, m)
\end{aligned}$$

where we identified the first term as the occurrence probability in the $(\ell - l)$-suffix and the second term as the length-mass distribution of the first fragment.

The same arguments also apply to the second recurrence. □

Most importantly, each of the recurrences in Theorem 9.2 uses only quantities of one particular mass m; we can therefore compute the occurrence probabilities independently for different masses.

The two recurrences in Theorem 9.2 are so-called full-history recurrences, i.e. all previous values $\bar{p}^{\langle \ell - l \rangle}(m)$ for $1 \leq l \leq \ell$ are needed for computing the value at index ℓ. In particular, for computing the ℓ-th value, we need to sum over ℓ terms, i.e. the number of operations increases with increasing string length ℓ.

Constant-order recurrences. Once more, the simple observation that a fragment can only have a certain mass m if its length is in a certain range allows us to modify the recurrences of Theorem 9.2 into recurrences of constant order that immediately lead to efficient implementations using dynamic programming techniques in Section 9.3.

Let us follow these modifications step-by-step to see what is going on before formally stating the result:

First, let us split the convolution sum's range into two parts up to and beginning after $l_{\max} + 1$:

$$\bar{p}^{\langle \ell \rangle}(m) \;=\; \sum_{l=1}^{l_{\max}+1} \bar{f}_1^{\langle \ell \rangle}(l, m) \cdot \bar{p}_+^{\langle \ell - l \rangle}(m) \;+\; \sum_{l=l_{\max}+2}^{\ell} u_1^{\langle \ell \rangle}(l) \cdot \bar{p}_+^{\langle \ell - l \rangle}(m).$$

The first part $1 \leq l \leq l_{\max} + 1$ covers the fragment lengths which may give fragments of mass $m \leq m_{\max}$. The second part $l_{\max} + 2 \leq l \leq \ell$ covers the fragment lengths for which mass m cannot be composed anymore. Thus, $\bar{f}_\circ^{\langle \ell \rangle}(l, m) = u_\circ^{\langle \ell \rangle}(l)$ for these lengths. Of course, we may also compute the maximal fragment length for each mass m separately, but for clarity of exposition we stick to $l_{\max}$, an upper bound valid for all masses $m \leq m_{\max}$.

The recurrence equations for fragment length distributions in Theorem 6.4 are only valid for the case of infinite string length. In order to apply them here, we first have to get rid of the $\langle \ell \rangle$-superscript. Since the distributions for the finite string length ℓ agree with the distributions for an infinite string up to $l = \ell - 1$ (cf. Lemma 6.13), we extract the ℓ-th term from the second sum:

$$\bar{p}^{\langle \ell \rangle}(m) = \sum_{l=1}^{l_{\max}+1} \bar{f}_1(l, m) \cdot \bar{p}_+^{\langle \ell - l \rangle}(m) \;+\; \sum_{l=l_{\max}+2}^{\ell-1} u_1(l) \cdot \bar{p}_+^{\langle \ell - l \rangle}(m) \;+\; u_1^{\langle \ell \rangle}(\ell).$$

For better readability, let us denote the first sum by $\Upsilon_1(\ell, m)$:

$$\Upsilon_1(\ell, m) \;:=\; \sum_{l=1}^{l_{\max}+1} \bar{f}_1(l, m) \cdot \bar{p}_+^{\langle \ell - l \rangle}(m),$$

and the second sum by $\Phi_1(\ell, m)$:

$$\Phi_1(\ell, m) \;:=\; \sum_{l=l_{\max}+2}^{\ell-1} u_1(l, m) \cdot \bar{p}_+^{\langle \ell - l \rangle}(m),$$

so we get

$$\bar{p}^{\langle \ell \rangle}(m) \;=\; u_1^{\langle \ell \rangle}(\ell) \;+\; \Upsilon_1(\ell, m) \;+\; \Phi_1(\ell, m)$$

The crucial step is to establish a recurrence equation for $\Phi_1(\ell, m)$. Not surprisingly, we can do this by using the recurrence equations for the length distributions of Theorem 6.4. These recurrence equations are linear and have constant coefficients; they almost immediately translate to recurrences on partial sums of their terms.

Lemma 9.3 (Recurrence equations for Φ_1 and Ψ). *Let*

$$\Phi_1(\ell, m) := \sum_{l=l_{\max}+2}^{\ell-1} u_1(l, m) \cdot \bar{p}_+^{\langle \ell - l \rangle}(m),$$
$$\Psi(\ell, m) := \sum_{l=l_{\max}+2}^{\ell-1} v(l, m) \cdot \bar{p}_+^{\langle \ell - l \rangle}(m).$$

With the notations of Theorem 6.4, the following recurrence equations hold for $\ell > l_{\max}$:

$$\Phi_1(\ell, m) = p_{\bar{\Gamma}} \cdot \Phi_1(\ell - 1, m) \;+\; p_\Gamma \cdot \Psi(\ell - 1, m) \;+\; u_1(l_{\max} + 2) \cdot \bar{p}^{\langle \ell - (l_{\max}+2) \rangle}(m),$$
$$\Psi(\ell, m) = p_{\Gamma \cap \Pi} \cdot \Phi_1(\ell - 1, m) \;+\; p_{\Gamma \cap \Pi} \cdot \Psi(\ell - 1, m) \;+\; v(l_{\max} + 2) \cdot \bar{p}^{\langle \ell - (l_{\max}+2) \rangle}(m).$$

Proof. Using Theorem 6.4,

$$\Phi_1(\ell, m) = p_{\bar{\Gamma}} \cdot \sum_{l=l_{\max}+2}^{\ell-1} u_1(l-1) \cdot \bar{p}_+^{\langle \ell - l \rangle}(m) \;+\; p_\Gamma \cdot \sum_{l=l_{\max}+2}^{\ell-1} v(l-1) \cdot \bar{p}_+^{\langle \ell - l \rangle}(m).$$

Performing an index shift in the two sums yields

$$\Phi_1(\ell, m) = p_{\bar{\Gamma}} \cdot \sum_{l=l_{\max}+1}^{\ell-2} u_1(l) \cdot \bar{p}_+^{\langle \ell - (l+2) \rangle}(m) \;+\; p_\Gamma \cdot \sum_{l=l_{\max}+1}^{\ell-2} v(l) \cdot \bar{p}_+^{\langle \ell - (l+2) \rangle}(m),$$

which, by letting the sums start at $l = l_{\max} + 2$ is

$$\begin{aligned}\Phi_1(\ell, m) = p_{\bar{\Gamma}} \cdot & \sum_{l=l_{\max}+2}^{\ell-2} u_1(l) \cdot \bar{p}_+^{\langle \ell - 1 - l) \rangle}(m) \\ + \; p_\Gamma \cdot & \sum_{l=l_{\max}+1}^{\ell-2} v(l) \cdot \bar{p}_+^{\langle \ell - 1 - l) \rangle}(m) \\ + \; & (p_{\bar{\Gamma}} u_1(l_{\max} + 1) \cdot + p_\Gamma v(l_{\max} + 1)) \cdot \bar{p}_+^{\langle L - 1 - (l_{\max}+1) \rangle}(m).\end{aligned}$$

We immediately identify the first sum as $\Phi_1(\ell-1,m)$ and the second sum as $\Psi(\ell-1,m)$. The last term, using Theorem 6.4 once more, yields $u_1(l_{\max}+2)\cdot\bar{p}^{\langle\ell-(l_{\max}+2)\rangle}$. The proof of the Ψ-recurrence is almost exactly the same and omitted here. □

Both recurrences are recurrences of order 1; to compute the values at index ℓ, we only need the two values at index $(\ell-1)$. The whole recurrence for $\bar{p}^{\langle\ell\rangle}(m)$ has then order $l_{\max}$, since $\Upsilon_1(\ell,m)$ involves the previous $l_{\max}$ values of the recurrence.

What about $\bar{p}_+^{\langle\ell\rangle}(m)$? This equation reads

$$\bar{p}_+^{\langle\ell\rangle}(m) \;=\; \sum_{l=1}^{l_{\max}+1} \bar{f}_+^{\langle\ell\rangle}(l,m)\cdot\bar{p}_+^{\langle\ell-l\rangle}(m) \;+\; \sum_{l=l_{\max}+2}^{\ell} u_+^{\langle\ell\rangle}(l)\cdot\bar{p}_+^{\langle\ell-l\rangle}(m).$$

Like above, let us define

$$\Upsilon_+(\ell,m) := \sum_{l=1}^{l_{\max}+1} \bar{f}_+^{\langle\ell\rangle}(l,m)\cdot\bar{p}_+^{\langle\ell-l\rangle}(m),$$
$$\Phi_+(\ell,m) := \sum_{l=l_{\max}+2}^{\ell-1} u_+^{\langle\ell\rangle}(l)\cdot\bar{p}_+^{\langle\ell-l\rangle}(m).$$

Next, let us take a very last glimpse at Theorem 6.4 to recall that we can write $u_+(l)$ in terms of $u_1(l-1)$ and $v(l-1)$. This leads immediately to the equation

$$\Phi_+(\ell,m) \;=\; \frac{p_{\bar{\Gamma}\cap\bar{\Pi}}}{p_{\bar{\Gamma}}}\cdot\Phi_1(\ell,m) \;+\; \frac{p_{\Gamma\cap\bar{\Pi}}}{p_{\bar{\Gamma}}}\cdot\Psi(\ell,m) \;+\; u_+(l_{\max}+2)\cdot\bar{p}_+^{\langle\ell-(l_{\max}+2)\rangle}(m),$$

so we can also apply the recurrences of Lemma 9.3 here.

We thus proved the following lemma.

Lemma 9.4 (Constant-order recurrences for $\mathbf{p}^{\langle\ell\rangle}$ and $\bar{\mathbf{p}}^{\langle\ell\rangle}$). *For a general cleavage scheme, the mass occurrence probabilities $p^{\langle\ell\rangle}$ and $\bar{p}_+^{\langle\ell\rangle}(m)$ can be computed for $\ell > l_{\max}$ by the recurrences*

$$\bar{p}^{\langle\ell\rangle}(m) = u_1^{\langle\ell\rangle}(\ell) \;+\; \Upsilon_1(\ell,m) \;+\; \Phi_1(\ell,m),$$
$$\bar{p}_+^{\langle\ell\rangle}(m) = u_+^{\langle\ell\rangle}(\ell) \;+\; \Upsilon_+(\ell,m) \;+\; \left(\frac{p_{\bar{\Gamma}\cap\bar{\Pi}}}{p_{\bar{\Gamma}}}\cdot\Phi_1(\ell,m) \;+\; \frac{p_{\Gamma\cap\bar{\Pi}}}{p_{\bar{\Gamma}}}\cdot\Psi(\ell,m) \;+\; u_+(l_{\max}+2)\cdot\bar{p}_+^{\langle\ell-(l_{\max}+2)\rangle}(m)\right)$$

with initial values for $\ell \le l_{\max}$ as given by Theorem 9.2. It is understood that quantities with negative indices are zero.

These two recurrences have order $l_{\max}+1$; the orders are constant in ℓ: For $\ell > l_{\max}$, the terms $\Upsilon_\circ(\ell,m)$ are each a summation of $l_{\max}$ terms; the number of terms is independent of the string length ℓ. Both $\Phi_1(\ell,m)$ and $\Psi(\ell,m)$ are given via a system of recurrences of order 1.

9.2. Approximation

In Section 9.3, we will use Lemma 9.4 to develop efficient exact algorithms for computing occurrence probabilities in the case of standard cleavage schemes. We will encounter two major problems: The computation time of the mentioned efficient algorithms and the more serious problem of memory requirements for keeping all occurrence probabilities in the main memory.

If we do not insist on computing each value exactly, we can also address both problems by an approximation of the values. Let us interpret the occurrence probabilities $p^{\langle \ell \rangle}(m)$ as values of a bivariate discrete function in ℓ and m. Since the combinatorial effects already encountered in the fragment length-mass and fragment mass distributions carry over to the occurrence probabilities (see Figure 9.3 for an illustration), we concentrate on finding reasonable approximations of the $m_{\max}$ functions $p^{\langle \ell \rangle}(\cdot)$ of string length. The occurrence probabilities for different masses are computed independently: We can thus give independent approximations for each fragment mass.

We distinguish the following four cases of behavior of $p^{\langle \ell \rangle}(m)$ taken as a function of string length ℓ for some fixed mass m. For illustration, examples are given for tryptic digestion and the amino acid alphabet without terminal characters.

1. **Mass m is only decomposable as the first fragment.** Then, $p^{\langle \ell \rangle}(m)$ decreases from $\lfloor m/\mu_{\max} \rfloor \leq \ell \leq \lceil m/\mu_{\min} \rceil$ and becomes constant for greater string lengths. It is zero outside this length-range if the mass m is not decomposable while obeying the composition rules for fragments.
   Example: $\mu('PK')$, as all fragments except the first begin with a non-prohibition character.

2. **Mass m is only decomposable as the last fragment.** The behavior is the same as in the previous case, $p^{\langle \ell \rangle}(m)$ becomes constant.
   Example: $\mu('L')$, as all fragments except the last must contain a cleavage character. Note that this mass is also a valid first fragment for string length $\ell = 1$.

3. **Mass m is decomposable as an inner fragment.** With increasing string length ℓ, chances increase that a fragment of mass m appears in the fragmentation: $p^{\langle \ell \rangle}(m)$ increases monotonically from some string length on. Note that every structure of an inner fragment is also a valid structure for the last fragment; this last fragment may simply end on a cleavage character by chance. The converse is not true, as the previous example shows.
   Example: $\mu('AK')$ is a valid first, inner and last fragment mass.

4. **Mass m is not decomposable as a fragment.** If mass m cannot be composed from character weights in such a way that the composition constraints for any fragment type are fulfilled, $p^{\langle \ell \rangle}(m) \equiv 0$ for all string lengths ℓ.
   Example: $\mu('KK')$ cannot be composed from any other character composition than two cleavage characters *'KK'*. This string is not a valid fragment, however, and can therefore not appear in a fragmentation.

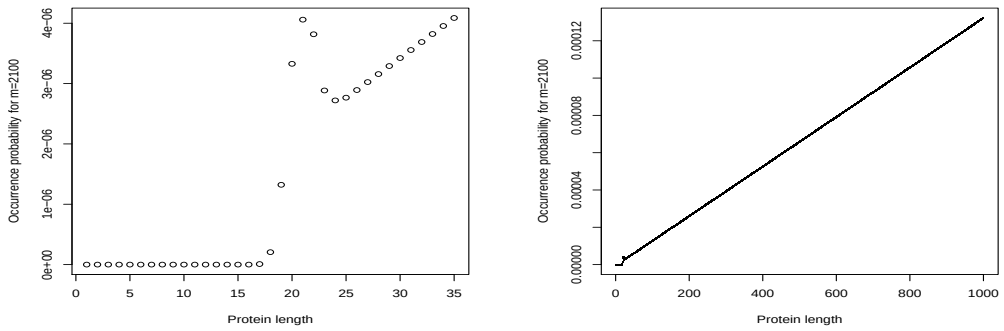

Figure 9.1.: Occurrence probabilities as function of protein length. Mass $m^* = 2\,100.0$ Da, 0.1 Da precision, tryptic digestion. Right: Length range $\ell = 1 \ldots 1\,000$. Left: Length range $\ell = 1 \ldots 35$.

The goal of this section is to investigate how the occurrence probabilities $p^\ell(m)$ can be written in parametric form as a function of ℓ, with parameters solely depending on m. We will have to take a little extra care for the boundaries: The first and the last fragment have fewer combinatorial constraints on their amino acid composition than the inner fragments. If the protein is short and does only have one fragment, there are even less combinatorial constraints. Let us have a look at Figure 9.1 (left) to see what consequences these different boundary cases have: The occurrence probability for $m = 21\,000$ (2 100.0 Da) first increases with string length, gets to a local maximum, decreases and finally increases monotonically. The mode in the beginning is due to the weaker constraints for small string length; there are simply more possibilities to build a string of mass m if the first and last character are arbitrary instead of a non-prohibition and a cleavage character, respectively. Nevertheless, the lengths of the first and last fragment can be expected to be fairly small, as the fragment length distribution is almost geometric for all types of fragments. From a certain string length on, the contribution of these boundary fragments to the mass occurrence probability can be expected constant; the increase in the mass occurrence probability is then determined solely by the inner fragments. Since the mass distributions of inner fragments do not have boundary effects and the inner fragments are i.i.d., we see a monotonic increase as in Figure 9.1 (right). For masses not decomposable as an inner fragment, the occurrence probabilities stay constant.

The main idea to approximate the occurrence probability of fragment mass m as a function of protein length ℓ is the following: This occurrence probability is the waiting time for the first occurrence of a fragment of mass m. From a certain protein length $\ell_{\min}$ on, the boundary effects stay constant and the increase is monotone. If we neglect the first and the last fragment, the occurrence probability can now be seen as the waiting time in a Bernoulli process where each fragment is seen as an element of the process. Each fragment mass either has or has not mass m. The waiting time for a fragment of

mass m can be expected to be geometric-like, i.e. of the form

$$\mathbb{P}(T^{\langle \ell \rangle}(m) \leq \ell) \;\approx\; 1 - c_m \cdot q_m^{\ell} \;=:\; \widehat{p^{\langle \ell \rangle}(m)}, \tag{9.1}$$

for $\ell > \ell_{\min}$. The parameters c_m and q_m are constants depending solely on m.

The parameter q_m describes the slope of the function. Given that the boundary effects can be neglected from $\ell_{\min}$ on, we can use the estimator

$$\widehat{q_m} \;=\; \sqrt[\Delta_\ell]{\frac{\bar{p}^{\langle \ell \rangle}(m)}{\bar{p}^{\langle \ell - \Delta_\ell \rangle}(m)}},$$

for some protein length difference Δ_ℓ and some protein length $\ell \geq \ell_{\min} + \Delta_\ell$. Choosing a greater length difference Δ_ℓ, gives a more accurate estimation of q_m. For our estimation, we took $\ell_{\min} = 500$ and $\Delta_\ell = 500$, estimating at $\ell = 1\,000$. However, an evaluation of estimators for $m = 2\,100.0$ Da at mass precision $\Delta_m = 0.1$ with $\ell_{\min} = 100$ and $\Delta_\ell = 1, 2, \ldots, 1\,000$ gave exactly the same estimates for each length difference. The estimation of q_m is therefore very robust even for small length differences and small starting protein length $\ell_{\min}$.

The approximation is only valid for protein lengths greater than $\ell_{\min}$ and we thus have to account for the value of the function at $\ell = \ell_{\min}$. This is done by introducing the parameter c_m that allows to set the value for the approximation at $\ell = \ell_{\min}$. We estimate this parameter by

$$\widehat{c_m} \;=\; \frac{\bar{p}^{\langle \ell_{\min} \rangle}(m)}{\widehat{q_m}^{\ell_{\min}}}.$$

The denominator is used for purely esthetic reasons: It allows us to take the ℓ-th power of q_m in Equation (9.1) to approximate $p^{\langle \ell \rangle}(m)$ instead of $p^{\langle \ell + \ell_{\min} \rangle}(m)$.

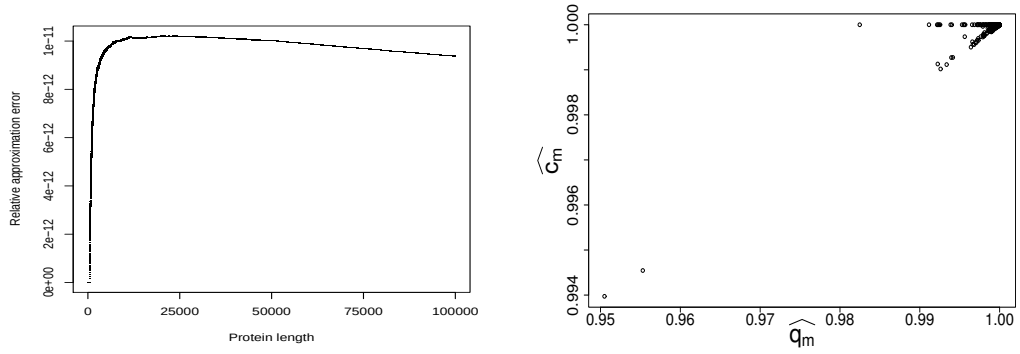

Figure 9.2.: Quality of mass occurrence probability approximations, $\Delta_m = 0.1$. Left: Relative error for fragment mass $m = 2\,100.0$ Da, protein lengths up to $\ell_{\max} = 100\,000$. Right: Initial parameter $\widehat{c_m}$ against slope parameter $\widehat{q_m}$ for $m = 2\,100.0$ Da.

We test the approximation for tryptic digestion, average amino acid masses and mass precision $\Delta_m = 0.1$ for each mass and length up to $\ell_{\max} = 10\,000$ and found the approximation errors within the computational accuracy of double precision for floating point arithmetic.

To see whether the estimation of the slope q_m is correct even for greater protein length, we computed the occurrence probability for mass $m = 2\,100.0$ Da with mass precision $\Delta_m = 0.1$, tryptic digestion and average amino acid masses. In Figure 9.2 (left), the relative approximation error

$$\varepsilon_{\text{rel}} := \frac{\left|\bar{p}^{\langle\ell\rangle}(m) - \widehat{\bar{p}^{\langle\ell\rangle}(m)}\right|}{\bar{p}^{\langle\ell\rangle}(m)}$$

is given for protein lengths up to $\ell_{\max} = 100\,000$; it stays within neglectable boundaries even for very large protein length. Note that for $\ell < \ell_{\min} = 500$, we have $\widehat{\bar{p}^{\langle\ell\rangle}(m)} = \bar{p}^{\langle\ell\rangle}(m)$.

In Figure 9.2 (right), the parameter estimate $\widehat{c_m}$ for the initial approximation at $\ell = \ell_{\min}$ is plotted against the slope parameter estimate $\widehat{q_m}$, both for fragment masses $57.0 \leq m \leq 3\,500.0$ and $\Delta_m = 0.1$. In this figure, we observe some of the previously mentioned points: In total, there are 34 430 different parameter pairs. For 1 089 of them, we observe $\widehat{c_m} = \widehat{q_m} = 1$ (they all gather in one point in the right upper corner of the figure). These masses are not decomposable as any type of fragment, and we have $p^{\langle\ell\rangle}(m) \equiv 0$ for all protein lengths. For another 4 789 pairs, we observe $\widehat{c_m} \neq 1$, but $\widehat{q_m} = 1$; these pairs make up the points on a horizontal line in the top of the figure. The corresponding masses are decomposable as a first or last fragment, i.e. the occurrence probability at $\ell_{\min}$ is greater than zero. However, they cannot be decomposed as an inner fragment and thus the occurrence probability stays constant for all subsequent protein lengths. An example is $m = 71.1$ Da, the average mass of amino acid 'A' that can only occur as the last fragment (for $\ell > 1$). All other parameter pairs have both $\widehat{c_m} \neq 1$ and $\widehat{q_m} \neq 1$. Not surprising, these pairs all lie around a straight line, i.e. they are highly correlated: Fragment masses with a high probability that can occur as both first and following fragments have a high initial probability and the occurrence probability also increases at a high rate. Last, the two points in the lower left corner of the figure correspond to the two cleavage character masses $\mu('K') = 128.2$ Da (left point) and $\mu('R') = 156.2$ Da (right point). They both have a high occurrence probability in the first part of a protein which also increases very fast to one for greater protein lengths.

9.3. Implementation

For computing the joint length-mass distributions of fragments in Section 8.5, we restrict the range of computation to those masses $m \leq m_{\max}$ that are detectable in a particular experimental setting. The same argument still holds for computing the occurrence probabilities: Since we assume that fragment masses greater than $m_{\max}$ are not detectable, we will only need the occurrence probabilities for masses up to $m_{\max}$.

In contrast to the length-mass distributions, however, we cannot restrict our computation to a small length range: We need to compute the occurrence probabilities up to some maximal protein length $\ell_{\max}$, usually defined by the longest protein in some reference database.

Time complexity. An implementation of the general recurrences in Theorem 9.2 using dynamic programming techniques has a time complexity of $\mathcal{O}(\ell)$ for computing the ℓ-th entry for any mass m, given the previous entries for this mass. Computing the whole table of occurrence probabilities up to some maximal fragment mass $m_{\max}$ and some maximal protein length $\ell_{\max}$ thus takes time $\mathcal{O}(\ell_{\max}^2 \cdot m_{\max})$.

The time complexity can be reduced considerably by implementing the recurrences given in Lemma 9.4 once the protein length ℓ exceeds $l_{\max}+1$. Computing the occurrence probabilities up to this length takes $\mathcal{O}(l_{\max}^2 \cdot m_{\max})$ time using the basic recurrences of Theorem 9.2.

The time complexity for computing the occurrence probability for one particular mass and protein length ℓ is $\mathcal{O}(l_{\max})$, given the values for previous lengths. The terms $\Phi_1(\ell - 1, m)$ and $\Psi(\ell - 1, m)$ are already computed and can be accessed in constant time $\mathcal{O}(1)$. From them, $\Phi_1(\ell, m)$ is computed in constant time using Lemma 9.3, as is $\Psi(\ell, m)$. For computing the terms $\Upsilon_\circ(\ell, m)$, we need to perform a convolution of size $l_{\max}$ for each term, given the joint length-mass distributions $f_\circ(\cdot, \cdot)$ for fragments; their time complexity is thus $\mathcal{O}(l_{\max})$.

Summarizing, the computations for the next protein length take $\mathcal{O}(l_{\max})$ time. The complexity for computing all necessary occurrence probabilities for fragment masses up to $m_{\max}$ and protein lengths up to $\ell_{\max}$ is therefore $\mathcal{O}(l_{\max} \cdot \ell_{\max} \cdot m_{\max})$. This complexity is linear in the maximal protein length compared to a quadratic complexity of the general implementation described above; the maximal fragment length is also usually very small.

Memory requirements. Whereas the maximal fragment length $l_{\max}$ defined by $m_{\max}$ is about 62 for a MALDI-TOF experiment, the maximal protein length in the Swiss-Prot sequence database is about 9 000. To guarantee constant and fast access times for the occurrence probabilities, we need to keep them in the main memory.

As a first demonstration on how much memory we actually need, let us consider a typical peptide mass fingerprint setting using a protein database. We would like to compute the occurrence probabilities $p^{\langle \ell \rangle}(\cdot)$ for tryptic digestion fragments up to, say, $m_{\max} = 3\,500$ Da with a precision of $\Delta_m = 0.1$ Da. For an identification using Swiss-Prot, we need to handle protein lengths up to $\ell_{\max} = 9\,000$. Assuming double precision for each probability, so 8 bytes per entry, we need $35\,000 \cdot 9\,000 \cdot 8$ bytes, about 2.35 GB of main memory just for the occurrence probabilities. For their computation, we also need to keep the length distributions and length-mass distributions, which requires another 35 MB of main memory (see Section 8.5). Moreover, we also need the $\bar{p}_+^{\langle \ell \rangle}(\cdot)$ entries in memory to compute the occurrence probabilities, which would nearly double the space requirements. These requirements are currently clearly out of question for contemporary desktop computers.

To resolve this issue, we first recall that for both $\bar{p}^{\langle\ell\rangle}(m)$ and $\bar{p}_+^{\langle\ell\rangle}(m)$, computation for different masses can be performed independently since only previously computed values for the same mass m are needed in the recurrences. We can compute $\bar{p}^{\langle\ell\rangle}(m)$ for some m up to $\ell_{\max}$ and immediately store $p^{\langle\ell\rangle}(m)$ for $\ell = 1 \dots \ell_{\max}$. The corresponding values of $\bar{p}_+^{\langle\ell\rangle}(\cdot)$ for mass m can be deleted since they were only needed to compute the corresponding values of $\bar{p}^{\langle\ell\rangle}(\cdot)$. Thus, we only need to keep two arrays of size $\ell_{\max}$ in memory to store $\bar{p}^{\langle\ell\rangle}(m)$ and $\bar{p}_+^{\langle\ell\rangle}(m)$ in addition to the $p^{\langle\ell\rangle}(m)$-values.

For computing the occurrence probabilities of mass m, we need the length-mass distributions $\bar{f}_\circ(\cdot,\cdot)$ only for the same mass m. We can therefore make use of our considerations in Section 8.5 and only keep a small part of these distributions for masses $m - \mu_{\max} \dots m$ in memory, computing the next entries from them when needed while removing other entries. This requires computing the occurrence probabilities in order of increasing mass, i.e. for $m, m+1, m+2, \dots$.

In summary, we need two extra arrays to keep $\bar{p}^{\langle\ell\rangle}(\cdot)$ and $\bar{p}_+^{\langle\ell\rangle}(\cdot)$ for the current mass, each of size $\ell_{\max}$, $3\,\mu_{\max}$ entries for the two length-mass distributions and $g(\cdot,\cdot)$, $2\ell_{\max}$ entries for the two length distributions and finally 2 entries for the previous $\Phi_1(\cdot, m)$ and $\Psi(\cdot, m)$ values.

The problem of keeping the occurrence probabilities in memory still remains serious. Clearly, it cannot be solved by smarter computation and we have to look for ways to reduce the memory requirements:

First, we expect some masses not to be decomposable, i.e. each entry in the corresponding column of $p^{\langle\ell\rangle}(\cdot)$ is zero. We may not want to store these entries. However, for the Swiss-Prot database and tryptic digestion with precision 0.1 Da, the number of such masses is negligible compared to the number of masses we need to store and compute. We would also need an additional data structure to keep track of mass indexing. Computing the above example, we found about 1 000 masses that were not decomposable for a mass range up to 35 000 (3 500 Da); these correspond to the masses for which the approximation parameters are $q_m = c_m = 1$, (cf. Section 9.2).

The other obvious possibility is to make use of our considerations for approximation in Section 9.2: If the occurrence probabilities depend smoothly on the protein length ℓ, we compute each entry for a particular mass m, but only store every D-th entry. We also need to store the first E values until $p^{\langle\ell\rangle}(m)$ becomes a smooth function in ℓ; we may choose $E \approx \ell_{\min}$. Then, only values for protein lengths $\ell = 1, 2, 3, \dots, E, E+D, E+2D, E+3D, \dots$ are kept in memory for each mass, resulting in a reduction of memory requirements of about a factor D for E small compared to $\ell_{\max}$. If an intermediate value between two nodes is accessed afterwards, it can be computed in constant time by a linear interpolation of the two nearest entries. Since E and D are known, these two entries can always be found in constant time.

For our example setting and the computations of occurrence probabilities in all following sections, we started storing interpolation nodes from $E = 100$ and stored every $D = 25$-th entry, reducing the memory requirement to about 100 MB. The interpolation error was below 10^{-10} for all entries.

9.4. Evaluation on Swiss-Prot

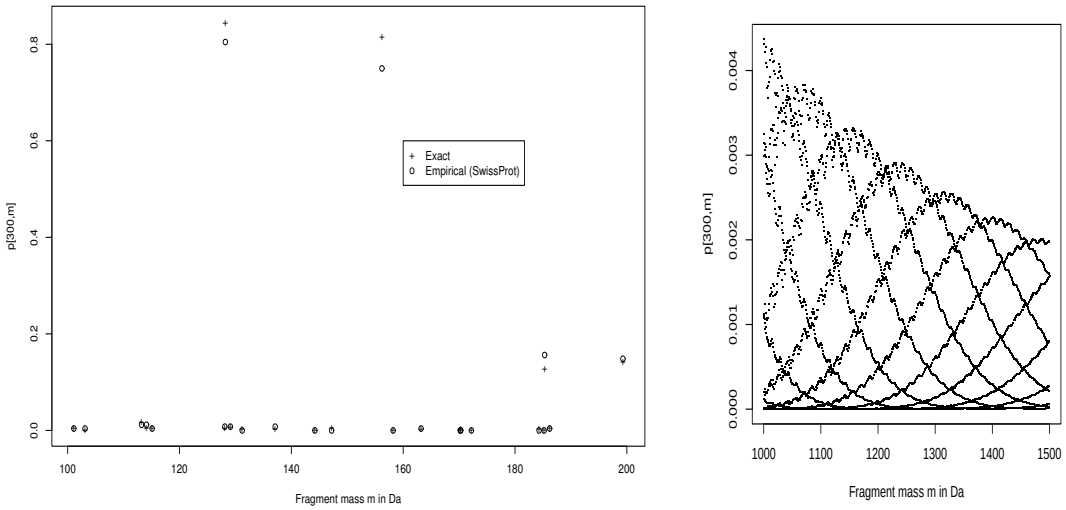

Figure 9.3.: Occurrence probabilities for fixed protein length $\ell = 300$, mass precision $\Delta_m = 0.1$, tryptic digestion. Left: Comparison to empirical Swiss-Prot data for masses $100 \leq m \leq 200$ Da, values above 10^{-4} shown. Right: Probabilities for masses $1\,000 \leq m \leq 2\,000$ Da.

We compare the mass occurrence probabilities predicted by our model with empirical frequencies of mass occurrences. To get a reasonably stable estimation of these frequencies for one protein length, we need a huge number of proteins of this particular length in the database so each fragment mass is covered by several fragments. We had to restrict our comparison to the comparably small mass range up to 200 Da; other fragment masses do not occur often enough to estimate their frequencies.

In Figure 9.3 (left), predicted occurrence probabilities in proteins of length $\ell = 300$ are shown as crosses together with their estimated frequencies as circles. The figure shows a reasonable agreement between model and data. The two most prominent probabilities of about ≈ 0.8 are given to the masses of the cleavage characters K and R. High probabilities (> 0.2) are also given to certain two-character fragments in this mass range. Note the similarity to the fragment mass distribution in Figure 8.3 in the corresponding mass range.

In Section 8.6, we observed a periodic behavior of the length-mass and mass distributions of fragments. A similar behavior is now observed in the occurrence probabilities for fixed protein length, see Figure 9.3 (right). Note that this figure shows *one* graph. As for the fragment length-mass distributions, this graph has period 20; taking the masses modulo this period "separates" the different "curves" visible in the graph. In Figure 9.4, the occurrence probabilities for masses $r = m \mod 20$ taken modulo 20 are shown for

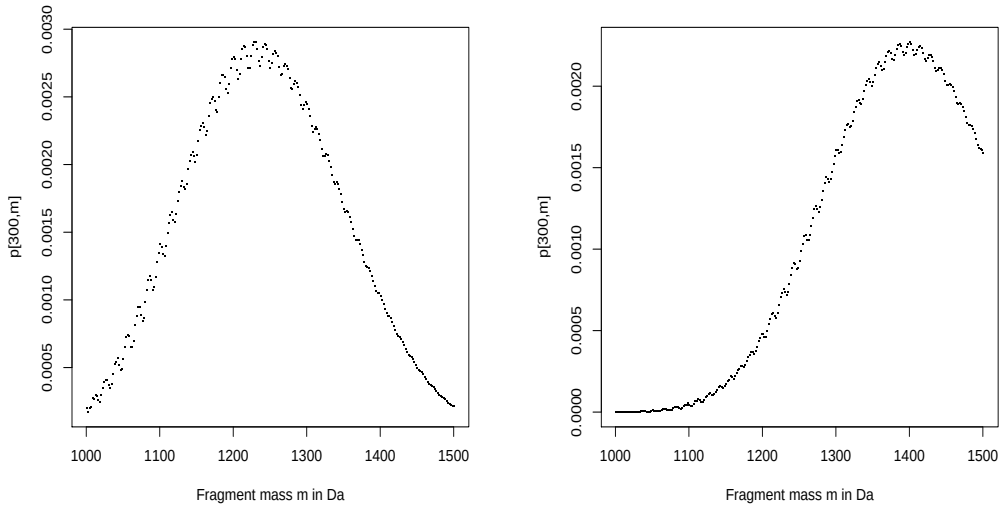

Figure 9.4.: Occurrence probabilities for fixed protein length $\ell = 300$, mass precision $\Delta_m = 0.1$, tryptic digestion, masses taken modulo 20: $r = m \mod 20$. Left: Remainder $r = 6$. Right: Remainder $r = 8$.

remainders $r = 6$ (left) and $r = 8$ (right).

9.5. Occurrence Probabilities for given Parent Mass

Instead of computing the occurrence probability of a certain fragment mass in a protein of some fixed length ℓ, we may want to compute this probability in a protein of given parent mass. This implies another statistical model for computing significance values. Just like in the case of finite fixed protein length, we assume that the parent mass is fixed and given from a source outside the probability model; we then compute conditional probabilities, given the fixed finite mass. In particular, this implies that we do not need a probability distribution for protein lengths or protein parent masses; it also implies that the sample space of considered strings is finite and each string has positive probability. The latter is not the case if strings of arbitrary lengths are considered. Then, each particular string and each finite set of strings would have probability zero. We might also introduce a probability distribution for protein lengths (e.g., estimated from Swiss-Prot), or a probability distribution for protein parent masses (e.g., directly estimated from Swiss-Prot or computed from a protein length distribution). This would allow us to compute the probability that a protein has a fragment certain mass *and* the protein itself has certain length or parent mass. This is a completely different model which we will not consider further.

Let us denote the occurrence probability of fragment mass m in a protein of parent

mass M by $p^{\langle M\rangle}(m)$ and the non-occurrence probability by $\bar{p}^{\langle M\rangle}(m)$. In particular, $p^{\langle M\rangle}(m)$ is the probability that a random weighted string has at least one fragment of mass m in its fragmentation, conditioned on the event that the string has a given mass.

We can now apply an argument similar to the one used for the case of given protein length: If a protein of mass M does not have a fragment of mass m, its first fragment must not have mass m and the remaining suffix after the first fragment must not contain a fragment of mass m. The mass of this suffix is the parent mass minus the mass of the first fragment. As before, we have to distinguish whether we take the first fragment of the whole protein or one of the suffixes remaining after a fragment. Sticking to the notation of Section 9.1, we denote the occurrence probability of fragment mass m in suffixes of mass M by $p_+^{\langle M\rangle}(m)$; non-occurrence probabilities are defined as $\bar{p}^{\langle M\rangle}(m) := 1-p^{\langle M\rangle}(m)$ and similar for suffixes.

In Section 8.4, we computed fragment mass distributions $f_\circ(m) = \mathbb{P}(\mu(F_\circ) = m)$ for first and following fragments in infinite strings. For finite parent mass, the protein length is of course also finite; the last fragment thus has a mass distribution different from the ones for a first or following fragment. Let $f_\circ^{\langle M\rangle}(m)$ denote the probability that a fragment has mass m if the parent mass of the protein is M. For reasons that will become clear later, we define

$$f_\circ^{\langle M\rangle}(m) \;:=\; \begin{cases} f_\circ(m), & \text{if } m \neq M, \\ \sum_{\ell=1}^{\infty} f_\circ^{\langle \ell\rangle}(\ell, M) & \text{if } m = M. \end{cases}$$

In this definition, fragment mass probabilities for masses greater than the parent mass are not zero. Note also that the probabilities do not form a probability distribution; they do not sum to one.

The case $m = M$ involves an infinite summation. This is not a real problem since for $\ell > \lceil M/\,\mu_{\min}\rceil$, all terms are zero and the number of non-zero summands is finite.

Theorem 9.5 (Mass occurrence probabilities for given parent mass). *For general cleavage schemes, the non-occurrence probability of fragment mass m in a random weighted string S of parent mass $\mu(S) = M$ is given by*

$$\bar{p}^{\langle M\rangle}(m) = \begin{cases} 1, & \text{if } m > M, \\ \sum_{m'\neq m} f_1^{\langle M\rangle}(m') \cdot \bar{p}_+^{\langle M-m'\rangle}(m), & \text{if } m \leq M, \end{cases}$$

where the non-occurrence probability in suffixes of mass M is

$$\bar{p}_+^{\langle M\rangle}(m) = \begin{cases} 1, & \text{if } m > M, \\ \sum_{m'\neq m} f_+^{\langle M\rangle}(m') \cdot \bar{p}_+^{\langle M-m'\rangle}(m), & \text{if } m \leq M. \end{cases}$$

Proof. For a protein of parent mass M not to have a fragment of mass m, its first fragment must not have mass m and the remaining suffix of mass $M - \mu(F_1)$ must not contain a fragment of mass m. In order to find the suffix' mass, we have to sum over all possible fragment masses for the first fragment. Care has to be taken if the fragment

mass equals the parent mass, as then a fragment of this mass is necessarily the protein's last (and only) fragment and we have to take the correct mass distribution $f_1^{\langle M \rangle}(M)$.

The same argument also applies to the occurrence probabilities in suffixes starting after a fragment. □

As in the case of given protein length, we can compute the occurrence probabilities for different fragment masses independently.

In principle, computing the occurrence probabilities using the recurrences of the previous theorem involves summation over an infinite mass range: It contains all masses except the fragment mass in question. However, it is possible to circumvent this problem by using the cumulative distribution function (cdf) of the fragment mass

$$G_\circ(m) \;:=\; \mathbb{P}\left(\mu\left(F_\circ\right) \leq m\right) \;=\; \sum_{m'=1}^{m} f_\circ(m').$$

We can re-derive the fragment mass probabilities from the cdf by the obvious relation $f_\circ(m) = G_\circ(m) - G_\circ(m-1)$.

Splitting the summation range in the recurrences into masses smaller, equal and greater than the parent mass yields

$$\bar{p}^{\langle M \rangle}(m) = \sum_{\substack{m' \neq m \\ m' < M}} f_1(m') \cdot \bar{p}_+^{\langle M-m' \rangle}(m) + f_1^{\langle M \rangle}(M) + \sum_{\substack{m' \neq m \\ m' > M}} f_1(m')$$

for $m \leq M$. The constraint $m' \neq m$ in the last sum is then superfluous and can be removed.

In order to decrease the computational effort for each recurrence step, we write the last sum in terms of the cdf $G_1(\cdot)$ as

$$\sum_{m' > M} f_1(m') \;=\; 1 - G_1(M),$$

and we may also write the first sum as

$$\sum_{\substack{m' \neq m \\ m' < M}} f_1(m') \cdot \bar{p}_+^{\langle M-m' \rangle}(m) \;=\; \sum_{\substack{m' \neq m \\ m' < M}} \big(G_1(m') - G_1(m'-1)\big) \cdot \bar{p}_+^{\langle M-m' \rangle}(m).$$

A similar argument also applies to the occurrence probabilities for suffixes.

We can compute the cdf's $G_\circ(\cdot)$ while computing the fragment mass distributions. Computing the M-th value for a certain mass, given the values for all previous parent masses involves a summation of $M+1$ terms. Thus, we can compute all occurrence probabilities up to some maximal parent mass $M_{\max}$ using standard dynamic programming techniques in time $\mathcal{O}(M_{\max}^2)$ for each fragment mass. In total, for parent masses up to $M_{\max}$ and fragment masses up to $m_{\max}$, we need time $\mathcal{O}(M_{\max}^2 \cdot m_{\max})$ and space $\mathcal{O}(M_{\max} \cdot m_{\max})$ to keep all entries. Both complexities are a serious problem in practice:

If we want to consider proteins up to length $\ell_{\max}$, we have an upper bound for the protein mass of $M_{\max} \leq \ell_{\max} \cdot \mu_{\max}$, a value much higher than $\ell_{\max}$, and increasing with increasing mass precision Δ_m.

We also need some additional space to store the relevant probabilities for computing the mass occurrence probabilities: We need $\mathcal{O}(m_{\max})$ space to keep the cdf for the fragment masses and $\mathcal{O}(M_{\max})$ space to keep the fragment mass probabilities for the boundary fragments.

Moreover, unlike for increasing protein length, we cannot expect the occurrence probabilities to depend smoothly on the parent mass; the probabilities not only depend on the decomposability of the fragment mass but also on the decomposability of the parent mass. If M is not decomposable, there is no protein of this parent mass and the occurrence probability is zero for any fragment mass.

Part II.

Protein Identification with Mass Spectra Alignments

10. Introduction

In the second part of the thesis, we develop a general computational framework for protein identification using peptide mass fingerprinting data. The framework is based on alignments of peak lists for computing similarity scores of a measured and a predicted spectrum. We use statistics developed in the first part to estimate the score distribution of peak list alignments and provide a p-value as statistical significance of an identification. We call this framework ***S**AMPI*[1]*: **A**ligning **M**ass spectra for **P**rotein **I**dentification.*

As we saw in Chapter 2.2, a raw spectrum is pre-processed into a peak list: First, signal processing algorithms are applied for noise reduction and baseline correction, then a peak-detection algorithm is applied for identification of the mass-to-charge ratios that correspond to measured ions, and finally the isotopic pattern and the charge state of these ions is determined for de-convolution and computing singly-charged peaks. The resulting peak list of m/z-values together with their intensities and possibly other attributes is taken as input for identification algorithms. Clearly, a lot of information might get lost or obscured by the variety of applied preprocessing methods and we have to keep this in mind when developing the identification framework.

Just to set the stage, let us briefly recall that measured peak lists are never perfect; they differ from what we expect from a theoretical peak list of a protein sequence: First, m/z-values are only indirectly measured, leading to potential shifts due to calibration and transformation errors. Second, peak lists contain additional peaks with arbitrary intensity, e.g. caused peak-detection errors and chemical noise. Third, some peaks may be missing due to peak-detection errors, probe preparation or insufficient ionization. Moreover, some peaks may have m/z-values close to each other and cannot be resolved by the MS machine, leading to a joint peak with m/z-value and intensity different from those of both ions.

Solely for readability, we limit our attention to ionization methods that predominantly produce single charged ions, such as MALDI. This allows us to talk about the mass of a molecule, instead of its mass-to-charge ratio.

We follow the general procedure common to all protein identification algorithms based on sequence database comparison: We compute the predicted peak list for each sequence in the sequence database using the same cleavage rules as the corresponding protease. This is done in a straightforward way using a mass table such as Table 1.1.

Each predicted peak list is then matched to the measured peak list to identify corresponding peaks. This matching is not unique and we need to select a matching that is optimal with respect to some criterion, measured by a similarity score. In Chapter 11, we give a general framework for efficiently computing the optimal matching of the pre-

[1] An ancient, obsolete greek character, cf. [70]

dicted and the measured peak list and reporting the matching score. The matching score serves as a numerical measure of how "close" the predicted spectrum is to the measured one, i.e. how good it explains the measurement. The highest scoring database sequence is then reported as the identification.

Since we allow quite general scoring schemes, a similarity score itself is only of limited use. In particular, it does not provide information about the quality of the identification. In Chapter 12, we therefore use the statistics of the first part of the thesis to compute a statistical significance of an identification under a well-defined null-model. Under some independence assumptions, the alignment score is shown to be nearly Gaussian; we estimate its parameters and compute a p-value for each computed alignment score.

In Chapter 13, we consider several aspects for constructing practical scoring schemes using mass differences and intensities. We evaluate our method on real-world proteomics data and compare our results with the standard software MASCOT.

11. Aligning Mass Spectra

11.1. Peaks and Peak Lists

Since we restrict ourselves to singly-charged ions, we can also restrict our attention to the mass of a peak; it is then identical with the mass-over-charge ratio. Modeling peaks and peak lists is then straightforward.

Definition 11.1 (Peak; peak list). A *peak list* of length n is an n-tuple $\mathcal{S} = (p_1, \ldots, p_n)$ of *peaks* $p_i \in \mathcal{M}^* \times \mathcal{A}$. Every such peak has a *mass* $\mu^*(p_i) \in \mathcal{M}^* \subseteq \mathbb{R}_{\geq 0}$ and possibly other additional attributes $(a_1(p_i), \ldots, a_k(p_i)) \in \mathcal{A}$, $k \geq 0$. A peak list is *ordered* with respect to mass, that is, $\mu^*(p_i) < \mu^*(p_j)$ whenever $i < j$ for all $1 \leq i, j \leq n$.

Note that we allow peaks without any attributes besides their mass. This gives the simplest representation of a peak. Note also that in contrast to the first part of the thesis, we are now working with peak masses given in Dalton if not explicitly stated otherwise, i.e. with real numbers (or, more precisely, floating point numbers in an implementation). To avoid confusion in later chapters, we denote quantities relating to real masses by a superscript $*$, whereas quantities without such superscript relate to scaled and rounded integer masses as introduced in the first part. In particular, $\mu^*(p)$ denotes a real non-negative peak mass, whereas $\mu(p)$ denotes a scaled and rounded non-negative integer mass of a peak p for some mass precision Δ_m. In the latter case, the set of peak masses $\mathcal{M}$ is a set of integers $\mathcal{M} = \{m_1, \ldots, m_u\}$ for some $u \in \mathbb{N}$. We always assume that the set of possible peak masses $\mathcal{M}^*$ and the mass precision Δ_m are given beforehand and are both constant.

Example 11.2 (Mass peak). The simplest representation of a peak is its mass. Then $\mathcal{M}^* = \mathbb{R}$ and $\mathcal{A} = \{\emptyset\}$.

The most important additional attribute of a peak is its intensity; it usually corresponds to the abundance of the molecule in the sample probe. Since intensity values also depend on a multitude of other parameters, such as ionization energy, or total amount of sample probe, it is common practice to use relative intensities, i.e. the highest intensity is set to one and all other intensity values in the peak list are scaled accordingly to a value between 0 and 1.

Example 11.3 (Peak with relative intensity). If we want to consider the relative intensity of a peak, we could set $\mathcal{M}^* = \mathbb{R}$ and $\mathcal{A} = [0, 1] \subset \mathbb{R}$. A peak p is then a pair $(\mu^*(p), a_1(p))$ of mass $\mu^*(p)$ and intensity $a_1(p)$.

Recall that depending on the experimental settings, there exists a maximal mass $m^*_{\max} \in \mathbb{R}_{\geq 0}$ such that no masses above $m^*_{\max}$ are present in any mass spectrum. For

example, $m^*_{\max} \approx 3\,500$ Da for tryptic digestion experiments on MALDI-TOF spectrometers. Then $\mathcal{M}^* := [0, m^*_{\max}]$ is the peak *mass range* of interest, and $l_{\max} := \lfloor m^*_{\max}/\mu^*_{\min} \rfloor$ is the maximal length of a fragment that we can detect; it is the same as in the first part of the thesis. The maximal mass corresponds to the scaled maximal fragment mass of the first part via the scaling $m_{\max} = \text{round}(m^*_{\max}/\Delta_m)$.

11.2. Peak List Matching and Scoring

Let $\mathcal{S}^{\text{p}} \equiv \mathcal{S}^{\text{p}}(s) = (p^{\text{p}}_1, \ldots, p^{\text{p}}_{n^{\text{p}}})$ be the predicted peak list of length n^{p} of some protein sequence s. We will only give the protein sequence of the predicted peak list if this sequence is of particular importance for the argument. Further, let $\mathcal{S}^{\text{m}} = (p^{\text{m}}_1, \ldots, p^{\text{m}}_{n^{\text{m}}})$ be a measured peak list of length n^{m}. We quantify the similarity of these two spectra by a numerical score in order to decide which predicted spectrum best explains the measurement.

We explicitly allow different attribute sets for the peaks in $\mathcal{S}^{\text{p}}$ and $\mathcal{S}^{\text{m}}$: Measured peaks usually have an intensity value, whereas peak intensities are not readily available in predicted peaks. We will come back to this in Chapter 13.

Matching peak lists. To quantify the similarity of two peak lists, we need to assign each peak in the measured peak list either to a peak in the predicted peak list or declare it as an additional peak if no such peak can be found. Conversely, we have to assign each peak in the predicted peak list either to a peak in the measured peak list or declare it as a missing peak.

Moreover, we want these peak assignments to preserve the relative order of the peak lists: Let $\mathcal{S}^{\text{m}}_\diamond \subseteq \mathcal{S}^{\text{m}}$ and $\mathcal{S}^{\text{p}}_\diamond \subseteq \mathcal{S}^{\text{p}}$ be two subsets of peaks that are matched to peaks in the respective other peak list, and let

$$\pi : \mathcal{S}^{\text{m}}_\diamond \to \mathcal{S}^{\text{p}}_\diamond$$

denote the assignment of a measured to a predicted peak. That is, $\pi(p^{\text{m}}_j) \in \mathcal{S}^{\text{p}}$ denotes the predicted peak assigned to the j-th measured peak. To preserve the relative order of the two peak lists in the assignment, the following condition must hold for any two peaks $p^{\text{m}}_j, p^{\text{m}}_{j'} \in \mathcal{S}^{\text{m}}_\diamond$:

$$\mu^*(\pi(p^{\text{m}}_j)) \leq \mu^*(\pi(p^{\text{m}}_{j'})) \text{ if and only if } \mu^*(p^{\text{m}}_j) \leq \mu^*(p^{\text{m}}_{j'}).$$

The same condition also holds for the corresponding two predicted peaks under π^{-1}. The sets $\mathcal{S}^{\text{m}}_\diamond$ and $\mathcal{S}^{\text{p}}_\diamond$ uniquely define the order-preserving matching π.

Restriction of the possible peak-matchings to the ones preserving the mass order of peaks is quite natural; the order of molecular mass of ions is also preserved in the MS measurement.

For the moment, let us assume that the assignment is one-to-one, i.e., π is a bijection. We will explain many-to-one assignments in Section 11.5.

Scoring peaks. We are not interested in all possible order-preserving peak assignments, but only in those that are optimal with respect to some optimality criterion. This criterion is defined by a peak scoring function *score* that gives a real value to each matched pair of peaks $(p^{\mathrm{p}}, p^{\mathrm{m}}) \in \mathcal{S}^{\mathrm{p}}_{\diamond} \times \mathcal{S}^{\mathrm{m}}_{\diamond}$. All unmatched peaks are assigned to a void peak.

Definition 11.4 (Peak scoring function). Let ε^{p} and ε^{m} denote special *gap peaks*. A *peak scoring function* or *scoring scheme*

$$score : \left(\mathcal{S}^{\mathrm{p}} \cup \{\varepsilon^{\mathrm{p}}\}\right) \times \left(\mathcal{S}^{\mathrm{m}} \cup \{\varepsilon^{\mathrm{m}}\}\right) \to \mathbb{R}$$

is then defined by

$$\begin{aligned} score\,(p^{\mathrm{p}}, p^{\mathrm{m}}) &= \Psi^{\mathrm{match}}\,(p^{\mathrm{p}}, p^{\mathrm{m}})\,, \\ score\,(\varepsilon^{\mathrm{p}}, p^{\mathrm{m}}) &= \Psi^{\mathrm{add}}\,(p^{\mathrm{m}})\,, \\ score\,(p^{\mathrm{p}}, \varepsilon^{\mathrm{m}}) &= \Psi^{\mathrm{miss}}\,(p^{\mathrm{p}})\,, \end{aligned}$$

and for completeness

$$score(\varepsilon^{\mathrm{p}}, \varepsilon^{\mathrm{m}}) = -\infty,$$

with the following *partial score functions*: The *matching score function*

$$\Psi^{\mathrm{match}} : (\mathcal{M}^* \times \mathcal{A}^{\mathrm{p}}) \times (\mathcal{M}^* \times \mathcal{A}^{\mathrm{m}}) \to \mathbb{R},$$

the *additional score function*

$$\Psi^{\mathrm{add}} : \mathcal{M}^* \times \mathcal{A}^{\mathrm{m}} \to \mathbb{R},$$

and the *missing score function*

$$\Psi^{\mathrm{miss}} : \mathcal{M}^* \times \mathcal{A}^{\mathrm{p}} \to \mathbb{R}.$$

Different gap peaks are needed to allow the two spectra to have different additional peak attributes. We do not specify gap peaks as mathematical objects, but rather define them implicitly via the partial scoring functions.

For the following considerations, we will refer to the peak scoring function $score(\cdot, \cdot)$, whereas the three partial scoring functions of a scoring scheme will be more appropriate for statistical computations in Chapter 12.

For $p^{\mathrm{p}} \in \mathcal{S}^{\mathrm{p}}$ and $p^{\mathrm{m}} \in \mathcal{S}^{\mathrm{m}}$, $score(p^{\mathrm{p}}, p^{\mathrm{m}})$ is the score of *matching peaks* p^{p} and p^{m}; $score(p^{\mathrm{p}}, \varepsilon^{\mathrm{m}})$ is the score of a *missing peak* p^{p} in $\mathcal{S}^{\mathrm{p}}$ not present in $\mathcal{S}^{\mathrm{m}}$ (e.g. by insufficient ionization); and $score(\varepsilon, p^{\mathrm{m}})$ is the score of an *additional peak* p^{m} in $\mathcal{S}^{\mathrm{m}}$ not present in $\mathcal{S}^{\mathrm{p}}$ (e.g. chemical noise).

It is clear that such a scoring function $score(\cdot, \cdot)$ must be based on the attributes of the peaks, such as mass or intensity: For example, if $\mu^*(p^{\mathrm{p}})$ is the mass of peak $p^{\mathrm{p}} \in \mathcal{S}^{\mathrm{p}}$ and $\mu^*(p^{\mathrm{m}})$ the mass of peak $p^{\mathrm{m}} \in \mathcal{S}^{\mathrm{m}}$, then $score(p^{\mathrm{p}}, p^{\mathrm{m}})$ should increase with decreasing

mass difference $|\mu^*(p^{\mathrm{p}}) - \mu^*(p^{\mathrm{m}})|$. The presented framework allows us to mimic additive or multiplicative scoring schemes, such as that used by MASCOT [103] or log likelihood peak scoring [38]. We will discuss some practical details of useful scoring schemes in Chapter 13.

We always require that a matching score is only positive for predicted peaks within a finite mass interval around a measured peak. We call this interval the *support* of the measured peak.

Definition 11.5 (Support of measured peak). The *support* $\mathcal{U}^*(p^{\mathrm{m}}) \subset \mathcal{M}^*$ of a measured peak is the interval of all peak masses that might give a positive matching score:

$$\mathcal{U}^*\left(p^{\mathrm{m}}\right) \;:=\; [m_l^*, m_r^*] \subset \mathbb{R}_{\geq 0},$$

where the left interval border is $m_l^* = \min_{m \in \mathcal{M}^*}\{\Psi^{\mathrm{match}}(p^{\mathrm{p}}, p^{\mathrm{m}}) > 0,\, \mu^*(p^{\mathrm{p}}) = m\}$ and similarly the right interval border is $m_r^* = \max_{m \in \mathcal{M}^*}\{\Psi^{\mathrm{match}}(p^{\mathrm{p}}, p^{\mathrm{m}}) > 0,\, \mu^*(p^{\mathrm{p}}) = m\}$.

Similarly, we define the discrete support for discrete scaled masses by the corresponding scaled quantities, given a mass precision Δ_m.

For discrete peak masses, the support is a set of consecutive integers: $\mathcal{U}(p^{\mathrm{m}}) = \{i, i+1, \ldots, i+n-1\} \subset \mathbb{N}_0$ for some i and n depending on the measured peak and the matching score function. We also require that supports of measured peaks are disjoint and we can always achieve this by shrinking support intervals.

Scoring matchings. The score of the bijective order-preserving matching $\pi : \mathcal{S}^{\mathrm{p}}_{\diamond} \to \mathcal{S}^{\mathrm{m}}_{\diamond}$ is the sum of scores of the peak matchings:

$$\mathit{score}(\pi) = \sum_{p^{\mathrm{m}} \in \mathcal{S}^{\mathrm{m}}_{\diamond}} \mathit{score}\big(\pi(p^{\mathrm{m}}), p^{\mathrm{m}}\big) + \sum_{p^{\mathrm{m}} \in \mathcal{S}^{\mathrm{m}} \setminus \mathcal{S}^{\mathrm{m}}_{\diamond}} \mathit{score}(\varepsilon^{\mathrm{p}}, p^{\mathrm{m}}) + \sum_{p^{\mathrm{p}} \in \mathcal{S}^{\mathrm{p}} \setminus \mathcal{S}^{\mathrm{p}}_{\diamond}} \mathit{score}(p^{\mathrm{p}}, \varepsilon^{\mathrm{m}}). \tag{11.1}$$

A matching is *optimal with respect to the peak scoring function*, if it has maximal score among all possible matchings.

An example. Let us take a brief look at a simple example, in order to get a little more intuition.

Example 11.6 (Peak-counting score). Using only peak masses for scoring, we define a *peak counting score* by

$$\mathit{score}(p^{\mathrm{p}}, p^{\mathrm{m}}) = \begin{cases} 1, & \text{if } \left|\mu^*(p^{\mathrm{p}}) - \mu^*(p^{\mathrm{m}})\right| \leq \delta \\ 0, & \text{otherwise} \end{cases}$$

for all $p^{\mathrm{p}} \in \mathcal{S}^{\mathrm{p}}$ and $p^{\mathrm{m}} \in \mathcal{S}^{\mathrm{m}}$ and for some fixed mass difference $\delta \in \mathbb{R}_{>0}$. The peak scores for unmatched peaks are set to zero: $\mathit{score}(p^{\mathrm{p}}, \varepsilon^{\mathrm{m}}) = \mathit{score}(\varepsilon^{\mathrm{p}}, p^{\mathrm{m}}) = 0$. Such a score counts the number of peaks we can match within a mass difference of at most δ.

11.3. Computing Optimal Matchings

So far, we only explained how a peak assignment of two peak lists is scored using peak scoring functions and gap peaks. We did not explain how an optimal assignment and its score are computed. From a bioinformatics perspective, this problem looks quite familiar: It is basically the same problem as a global sequence alignment with gaps. Global sequence alignments were introduced in [95] as a tool for the analysis of biological sequences. They are now a standard method for computing sequence similarities. They were also successfully applied to a variety of other problems such as *time warping* [117], physical map comparison [67], aligning gel electrophoresis patterns [3, 64] or matching tree ring data [128].

These algorithms allow the simultaneous computation of an optimal alignment together with its similarity score. If we interpret a peak as a letter in some (possibly infinite) alphabet, we can use the exact same methods for simultaneously computing an optimal peak list alignment together with its similarity score. The major difference between ordinary global sequence alignment and alignment of two peak lists is the structure of the scoring functions. Unlike sequence alignment, peak list alignment is not based upon an evolutionary model of substitution and insertion/deletion of characters (or peaks, in this case). Therefore, the concept of mismatches is completely absent in peak list scoring and it seems unnecessary to allow affine gap penalties.

We would like to stress that there is no correspondence between peak list alignments as introduced below and "spectral alignments" introduced in [107].

We formalize the above considerations.

Theorem 11.7 (Computing optimal peak list alignments). *Let $\mathcal{S}^p$ be a predicted peak list of size n^p, and let $\mathcal{S}^m$ be a measured peak list of size n^m. Moreover, let E be a $(n^p+1)\times(n^m+1)$ dynamic programming matrix for global alignment, and let $score(\cdot,\cdot)$ denote a peak-wise scoring function. Then the optimal alignment score of the two peak lists $\mathcal{S}^p$ and $\mathcal{S}^m$ can be computed in $\mathcal{O}(n^p \cdot n^m)$ time by the recurrence*

$$
\begin{aligned}
E(0,0) &= 0 \\
E(i+1,0) &= E(i,0) + score(p^p_{i+1}, \varepsilon^m) \\
E(0,j+1) &= E(0,j) + score(\varepsilon^p, p^m_{j+1}) \\
E(i+1,j+1) &= \max \left\{ \begin{array}{l} E(i,j+1) + score(p^p_{i+1}, \varepsilon^m), \\ E(i+1,j) + score(\varepsilon^p, p^m_{j+1}), \\ E(i,j) + score(p^p_{i+1}, p^m_{j+1}) \end{array} \right\}
\end{aligned}
$$

for all i with $0 \le i \le n^p$ and all j with $0 \le j \le n^m$. The optimal alignment score *of $\mathcal{S}^p$ and $\mathcal{S}^m$ is then*

$$score(\mathcal{S}^p, \mathcal{S}^m) \;:=\; \max_{\pi} \{score(\pi)\} \;=\; E(n^p, n^m),$$

where the maximum is taken over all possible order-preserving matchings π of the two peak lists. We can find all such optimal alignments by backtracing through the matrix E.

Recall that each order-preserving matching is uniquely defined by the two sets $\mathcal{S}^{\mathrm{m}}_{\diamond}$ and $\mathcal{S}^{\mathrm{p}}_{\diamond}$ of matched peaks, i.e., we also implicitly maximize taking into account all feasible sets of matched peaks. The sets of matched peaks for the highest scoring peak list matching are computed simultaneously; they correspond to the peak list alignment.

It should be understood that for reasonable peak scorings, we do not have to compute the complete matrix E: We can expect that $\mathit{score}(p^{\mathrm{p}}, p^{\mathrm{m}})$ decreases as the mass difference $|\mu^*(p^{\mathrm{p}}) - \mu^*(p^{\mathrm{m}})|$ increases. This behavior is guaranteed by the finite support of each measured peak. In particular, $\mathit{score}(p^{\mathrm{p}}, p^{\mathrm{m}})$ will be very small for high mass differences, because there is no reason to match two peaks that are, say, 1 000 Da apart. On the other hand, scores $\mathit{score}(p^{\mathrm{p}}, \varepsilon^{\mathrm{m}})$ and $\mathit{score}(\varepsilon^{\mathrm{p}}, p^{\mathrm{m}})$ are mostly independent of peak masses. Let θ be a lower bound of $\mathit{score}(p^{\mathrm{p}}, \varepsilon^{\mathrm{m}})$ and $\mathit{score}(\varepsilon^{\mathrm{p}}, p^{\mathrm{m}})$. From the above, we may assume that there exists some mass difference δ such that $\mathit{score}(p^{\mathrm{p}}, p^{\mathrm{m}}) \leq 2\theta$ for all peaks with $|\mu^*(p^{\mathrm{p}}) - \mu^*(p^{\mathrm{m}})| \geq \delta$. So, it suffices to compute only those parts of the matrix E where $|\mu^*(p^{\mathrm{p}}) - \mu^*(p^{\mathrm{m}})|$ is not too large. The optimal alignment can then be computed by "banded" dynamic programming in time $\mathcal{O}(|C| + |\mathcal{S}^{\mathrm{p}}| + |\mathcal{S}^{\mathrm{m}}|)$ where $C := \left\{(i,j) : \left|\mu^*(p^{\mathrm{p}}_i) - \mu^*(p^{\mathrm{m}}_j)\right| \leq \delta\right\}$ is the set of potential matches: For every peak p^{p}_i there exist indices l, r such that $\left|\mu^*(p^{\mathrm{p}}_i) - \mu^*(p^{\mathrm{m}}_j)\right| \leq \delta$ if and only if $j \in \{l, l+1, l+2, \ldots, r\}$. Going from i to $i+1$, we only have to increase the pointers l, r.

11.4. Examples

Example 11.8 (A simple scoring function). Given two spectra $\mathcal{S}^{\mathrm{p}} := \{p^{\mathrm{p}}_1, \ldots, p^{\mathrm{p}}_4\}$ and $\mathcal{S}^{\mathrm{m}} := \{p^{\mathrm{m}}_1, \ldots, p^{\mathrm{m}}_5\}$, let $\mu^*(p^{\mathrm{p}}_i) = 200, 510, 705, 850$, and let $\mu^*(p^{\mathrm{m}}_j) = 200, 300, 500, 515, 700$. For $\delta = 10$ and the "peak counting score" introduced in Example 11.6, we easily compute $E(4,5) = 3$, so an optimal alignment matches three peaks. The alignment matrix E then reads:

$E(i,j)$	ε^{m}	200	300	500	515	700
ε^{p}	0	0	0	0	0	0
200	0	1	1	1	1	1
510	0	1	1	2	2	2
705	0	1	1	2	2	3
850	0	1	1	2	2	3

For readability, we print masses $\mu^*(p^{\mathrm{p}}_i)$ and $\mu^*(p^{\mathrm{m}}_j)$ instead of indices i and j in these tables.

Example 11.9 (A more complex scoring function). For the same peak lists as in Example 11.8 and the slightly more complex peak scoring function

$$\begin{aligned}
&\mathit{score}(p^{\mathrm{p}}, p^{\mathrm{m}}) = 2 - \tfrac{1}{5}\left|\mu^*(p^{\mathrm{p}}) - \mu^*(p^{\mathrm{m}})\right|,\\
&\mathit{score}(p^{\mathrm{p}}, \varepsilon^{\mathrm{m}}) = -1, \text{ and}\\
&\mathit{score}(\varepsilon^{\mathrm{p}}, p^{\mathrm{m}}) = -1,
\end{aligned}$$

the alignment matrix E is:

$E(i,j)$	ε^{m}	200	300	500	515	700
ε^{p}	**0**	−1	−2	−3	−4	−5
200	−1	**2**	**1**	**0**	−1	−2
510	−2	1	0	1	**1**	0
705	−3	0	−1	0	0	**2**
850	−4	−1	−2	−1	−1	**1**

We have grayed out those entries of $E(i,j)$ that need not to be computed. So, an optimal alignment has score $E(4,5) = 1$; we can achieve this score by matching p_1^{p}(of mass 200) with $p_1^{\mathrm{m}}(200)$, $p_2^{\mathrm{p}}(510)$ with $p_4^{\mathrm{m}}(515)$, and $p_3^{\mathrm{p}}(705)$ with $p_5^{\mathrm{m}}(700)$.

11.5. Many-to-One Peak Matching

Often, we want to match a single measured sample peak p^{m} to one *or more* predicted reference peaks. The simplest incorporation of such many-to-one peak matchings is as follows. We add scores of matching a measured peak p^{m} to all predicted peaks p^{p} with mass $\mu^*(p^{\mathrm{p}}) \in \mathcal{U}^*(p^{\mathrm{m}})$, and if there is no such predicted peak, we score peak p^{m} by $\Psi^{\mathrm{add}}(p^{\mathrm{m}})$. Now, for a measured peak list $\mathcal{S}^{\mathrm{m}}$ and a predicted peak list $\mathcal{S}^{\mathrm{p}}$, the many-to-one alignment score is given by

$$\mathit{score}\,(\mathcal{S}^{\mathrm{p}}, \mathcal{S}^{\mathrm{m}}) := \sum_{p^{\mathrm{m}} \in \mathcal{S}^{\mathrm{m}}} \sum_{\substack{p^{\mathrm{p}} \in \mathcal{S}^{\mathrm{p}} \\ \mu^*(p^{\mathrm{p}}) \in \mathcal{U}^*(p^{\mathrm{m}})}} \Psi^{\mathrm{match}}\,(p^{\mathrm{p}}, p^{\mathrm{m}}) + \sum_{p^{\mathrm{m}}\ \text{add.}} \Psi^{\mathrm{add}}\,(p^{\mathrm{m}}) + \sum_{p^{\mathrm{p}}\ \text{miss.}} \Psi^{\mathrm{miss}}\,(p^{\mathrm{p}}) \tag{11.2}$$

where "p^{m} add." runs over those $p^{\mathrm{m}} \in \mathcal{S}^{\mathrm{m}}$ for which there is no $p^{\mathrm{p}} \in \mathcal{S}^{\mathrm{p}}$ with $\mu^*(p^{\mathrm{p}}) \in \mathcal{U}^*(p^{\mathrm{m}})$; "$p^{\mathrm{p}}$ miss." runs over those $p^{\mathrm{p}} \in \mathcal{S}^{\mathrm{p}}$ for which there is no $p^{\mathrm{m}} \in \mathcal{S}^{\mathrm{m}}$ with $\mu^*(p^{\mathrm{p}}) \in \mathcal{U}^*(p^{\mathrm{m}})$. We can compute *score* in time $\mathcal{O}(|\mathcal{S}^{\mathrm{p}}| \cdot |\mathcal{S}^{\mathrm{m}}|)$, or $\mathcal{O}(|C| + |\mathcal{S}^{\mathrm{p}}| + |\mathcal{S}^{\mathrm{m}}|)$ where C is again the set of potential matches.

12. Computing Significance of Alignment Scores

Using peak list alignments and their scores as introduced above allows us to select a best-scoring spectrum from, say, a database of sequences. However, recall that the alignment score itself is only of limited value: Longer protein sequences usually have a larger number of predicted peaks and thus a higher chance to give high scores than shorter sequences with less peaks. Moreover, the alignment score gives no information about the statistical significance of the alignment. To provide such a statistical significance, we have to answer the following question:

What is the probability that a certain score is achieved "just by chance"?

This question can be answered if we compute a p-value for each alignment score, which gives the probability that a score of a certain value or higher is achieved just by chance. For computing such p-value, we need two things: First, we need a proper null-model that exactly defines what "by chance" actually means. Second, we need the distribution of the alignment score under this null-model.

The alignment score depends on several factors: The chosen scoring scheme $score(\cdot,\cdot)$, the number of peaks in the measured peak list $\mathcal{S}^{\mathrm{m}}$ and the number of peaks in the predicted peak list $\mathcal{S}^{\mathrm{p}}$. For a particular identification, the scoring scheme is fixed and the measured peak list is given. A sensible null-model would thus define a probability distribution on the predicted peak lists, i.e. define random predicted peak lists. Here, the first part of the thesis enters the stage: Each finite random string $S^{\langle\ell\rangle}$ defines a random peak list $\mathcal{S}^{\mathrm{p}}(S^{\langle\ell\rangle})$. A random model on $S^{\langle\ell\rangle}$ thus defines a random model on predicted peak lists. Note that the correspondence between peak lists and strings is not one-to-one: Both strings $s = MPM$ and $t = PMM$ give rise to the same peak list $\mathcal{S}^{\mathrm{p}} = \{p_1^{\mathrm{p}}\}$ with $\mu(p_1^{\mathrm{p}}) = \mu(s) = \mu(t)$.

Instead of defining a random model on strings of certain length ℓ, we might also define such a model on strings with certain parent mass M.

The choice of either string length or parent mass defines two different types of null-models. If we choose the parent mass as parameter for the null-model, we use *one* null-model for all alignments of predicted peak lists to the measured peak list. Thus, p-values are comparable since they refer to the same alignment score distribution. Moreover, the parent mass is a parameter gained from the experiment; it might be known from a previous mass spectrometry run of the intact protein before digesting or it might be estimated from a 2D-gel. This requires more experimental effort: Additional MS runs on the intact protein require a larger amount of sample, and 2D-gels provide only a

rough estimate of the parent mass. In contrast, choosing string length as parameter for the null-model requires no additional measurement and there are no uncertainties in the parameter's value. However, this choice gives rise to *several* null-models (one model per sequence length), so p-values of different sequences are computed by different alignment score distributions.

We restrict our attention to fixed string length, since we developed more efficient algorithms for computing the mass occurrence probabilities. However, the following considerations remain exactly the same, regardless of our choice; we can replace each $\langle \ell \rangle$-superscript by a $\langle M \rangle$-superscript if we choose to take the parent mass for our null-model.

Now that we defined a null-model, what is the alignment score distribution under this null-model? Alignment score distributions of sequence alignments with and without gaps have been intensively studied and a vast amount of literature exists (see [126, 127] for some introduction). The most challenging problem is the optimality of an alignment score: We do not simply sum independent random variables (the scores of two aligned random characters), but we take the maximum of all possible choices, so the score of the optimal alignment is the maximal score of all possible alignments. This gives rise to extreme-value distributions, mostly derivates of the Gumbel-distribution with cumulative distribution function

$$F(x) \;=\; 1 - \exp(-a \cdot \exp(b \cdot x)).$$

The problem is now to efficiently compute the parameters a and b.

We are facing the same kind of problem here: We need to compute the peak list alignment score distribution, where this optimal alignment score is again the maximal score of all possible alignment scores. However, there is a substantial difference between peak list alignments and sequence alignments: Due to the finite support of measured peaks, the alignment of peak lists is much more local in the sense that we do not have to consider the whole row or column in the alignment matrix but rather a small band around the considered peak. Details were already given in Section 11.3. We can therefore assume that the alignment score distribution is almost the distribution of a sum of independent random variables. Then, a weaker version of the Central Limit Theorem applies and the alignment score distribution is Gaussian.

Independence assumption (I)
Hitherto, we assume that peak masses are mutually independent; the alignment score is then a sum of independent random variables and its distribution is approximately Gaussian.

We will provide numerical simulation results in Section 12.4 to validate this assumption in practical computations. We will also provide numerical simulation data for the scoring schemes used in an evaluation study in Chapter 13.

A Gaussian distribution is completely defined by its two parameters expectation and variance. Thus, we only need to estimate these two parameters given a scoring scheme, a measured spectrum and one of the two null-models. Figure 12.1 gives an informal visualization of the difference in the alignment matrices for sequence and peak list alignment,

where the shaded areas indicate feasible matrix entries for optimal alignments and two possible alignment paths are given in each matrix.

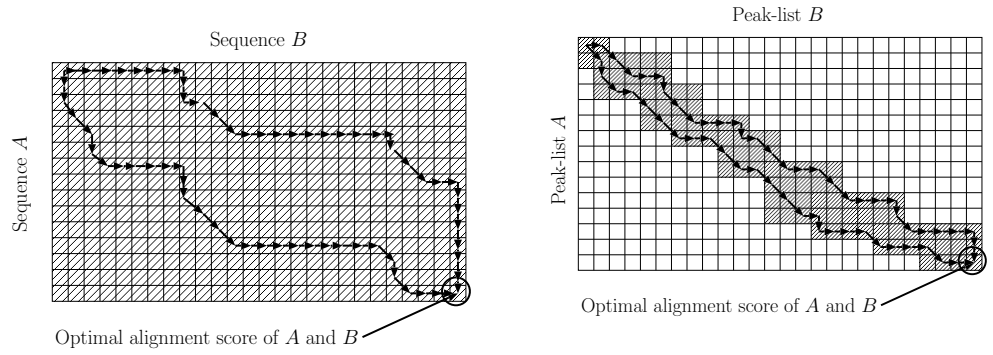

Figure 12.1.: Areas of possible optimal alignments paths (shaded) in an alignment matrix. Left: For sequence alignment, the path may be anywhere in the matrix. Right: For peak list alignment, the path is within a small band.

12.1. Moments of Alignment Scores

The alignment score $score(\mathcal{S}^{\mathrm{p}}, \mathcal{S}^{\mathrm{m}})$ of two peak lists $\mathcal{S}^{\mathrm{p}}$ and $\mathcal{S}^{\mathrm{m}}$ is the sum of the alignment scores of matched peaks, additional peaks, and missing peaks. Assuming independence of these peaks, the expectation and variance of the alignment score are thus the sum of the expectations and variances of the matched, additional, and missing peak-scores in the alignment. We will analyze the many-to-one peak matching scenario, because it allows us to model alignment scores using only mild independence assumptions. Let $A = score(\mathcal{S}^{\mathrm{p}}, \mathcal{S}^{\mathrm{m}})$ be the score of a given measured peak list aligned to a predicted peak list randomly chosen according to an appropriate null-model. Then,

$$\mathbb{E}(A) \;=\; \mathbb{E}\left(\sum_{p^{\mathrm{m}} \in \mathcal{S}^{\mathrm{m}}} \sum_{\substack{p^{\mathrm{p}} \in \mathcal{S}^{\mathrm{p}} \\ \mu^*(p^{\mathrm{p}}) \in \mathcal{U}^*(p^{\mathrm{m}})}} \Psi^{\mathrm{match}}(p^{\mathrm{p}}, p^{\mathrm{m}}) + \sum_{p^{\mathrm{m}}\ \mathrm{add.}} \Psi^{\mathrm{add}}(p^{\mathrm{m}}) + \sum_{p^{\mathrm{p}}\ \mathrm{miss.}} \Psi^{\mathrm{miss}}(p^{\mathrm{p}}) \right).$$

Further, we assume that predicted peaks do not have additional attributes besides their mass. For more complex peak types, probability distributions of the additional attributes have to be incorporated into the null-model.

The expected score of a measured peak list and a random predicted peak list is the sum of three expectations for the score of matched, additional, and missing peaks. Let us compute these expectations in order, before stating the alignment score expectation and variance. Recall that quantities involving masses are defined for real masses if they have a $*$ superscript and are defined for integer masses without it. We always assume that some mass precision Δ_m is chosen beforehand.

Lemma 12.1 (Moments for matched peaks). *Let $\mathcal{S}^m$ be a measured peak list, let score$(\cdot,\cdot)$ be a scoring scheme with matching score function $\Psi^{match}(\cdot,\cdot)$, and let the null-model be defined for finite string length ℓ.*

Let further A^{match} denote the partial alignment score of the matched peaks of $\mathcal{S}^m$ to a random peak list $\mathcal{S}^p$, under the null-model, i.e.

$$A^{match} := \sum_{p^m \in \mathcal{S}^m} \sum_{\substack{p^p \in \mathcal{S}^p \\ \mu^*(p^p) \in \mathcal{U}^*(p^m)}} \Psi^{match}\left(p^p, p^m\right).$$

Then, the k-th moment of the partial score for matched peaks is

$$\mathbb{E}\left(\left(A^{match}\right)^k\right) \approx \sum_{p^m \in \mathcal{S}^m} \sum_{m \in \mathcal{U}(p^m)} \Psi^{match}(m \cdot \Delta_m, p^m)^k \cdot p^{\langle \ell \rangle}(m),$$

where the approximation depends on the chosen mass precision Δ_m of the null-model.

Proof. The measured peak list $\mathcal{S}^{\mathrm{m}}$ is given and non-random. We can therefore extract the summation over this peak list from the expectation:

$$\begin{aligned}\mathbb{E}(A^{\mathrm{match}}) &= \mathbb{E}\left(\sum_{p^{\mathrm{m}} \in \mathcal{S}^{\mathrm{m}}} \sum_{\substack{p^{\mathrm{p}} \in \mathcal{S}^{\mathrm{p}} \\ \mu^*(p^{\mathrm{p}}) \in \mathcal{U}^*(p^{\mathrm{m}})}} \Psi^{\mathrm{match}}\left(p^{\mathrm{p}}, p^{\mathrm{m}}\right)\right) \\ &= \sum_{p^{\mathrm{m}} \in \mathcal{S}^{\mathrm{m}}} \mathbb{E}\left(\sum_{\substack{p^{\mathrm{p}} \in \mathcal{S}^{\mathrm{p}} \\ \mu^*(p^{\mathrm{p}}) \in \mathcal{U}^*(p^{\mathrm{m}})}} \Psi^{\mathrm{match}}\left(p^{\mathrm{p}}, p^{\mathrm{m}}\right)\right).\end{aligned}$$

We can re-write the conditions on the second sum by using the indicator function and switch over to integer masses. Since $\mathcal{S}^{\mathrm{p}}$ is random, we assumed peak to be independent, and peaks within a peak list are unique, we can introduce a peak P^{p}, randomly chosen from any random predicted peak list:

$$\mathbb{E}(A^{\mathrm{match}}) \approx \sum_{p^{\mathrm{m}} \in \mathcal{S}^{\mathrm{m}}} \mathbb{E}\left(\sum_{m \in \mathcal{U}(p^{\mathrm{m}})} \Psi^{\mathrm{match}}\left(P^{\mathrm{p}}, p^{\mathrm{m}}\right) \cdot \mathbb{1}_{\{\mu(P^{\mathrm{p}})=m\}}\right),$$

where the summation over peak-mass m is non-random and the scoring function uses mass as only attribute of the predicted peak. Note that the scoring function is only defined for real masses, so we need to re-scale the integer mass by multiplication with the mass precision Δ_m:

$$\mathbb{E}(A^{\mathrm{match}}) \approx \sum_{p^{\mathrm{m}} \in \mathcal{S}^{\mathrm{m}}} \sum_{m \in \mathcal{U}(p^{\mathrm{m}})} \mathbb{E}\left(\Psi^{\mathrm{match}}\left(m \cdot \Delta_m, p^{\mathrm{m}}\right) \mathbb{1}_{\{\mu(P^{\mathrm{p}})=m\}}\right).$$

Since the matching score is now non-random, we can extract it to get

$$\mathbb{E}(A^{\text{match}}) \approx \sum_{p^{\text{m}} \in \mathcal{S}^{\text{m}}} \sum_{m \in \mathcal{U}(p^{\text{m}})} \Psi^{\text{match}}\left(m \cdot \Delta_m, p^{\text{m}}\right) \cdot \mathbb{E}\left(\mathbb{1}_{\{\mu(P^{\text{p}})=m\}}\right).$$

Finally, the expectation of an indicator function is the probability of the corresponding event. Moreover, in the proposed null-model, the probability $\mathbb{P}(\mu(P^{\text{p}}) = m)$ is the occurrence probability of mass m in a random weighted string of length ℓ, thus

$$\mathbb{E}(A^{\text{match}}) \approx \sum_{p^{\text{m}} \in \mathcal{S}^{\text{m}}} \sum_{m \in \mathcal{U}(p^{\text{m}})} \Psi^{\text{match}}\left(m \cdot \Delta_m, p^{\text{m}}\right) \cdot p^{\langle \ell \rangle}(m),$$

as claimed. Higher moments are derived similarly.

□

Similarly, we compute the moments of additional peaks. Again, the approximation quality depends on the chosen mass precision Δ_m.

Lemma 12.2 (Moments of additional peaks). *Under the conditions of Lemma 12.1, let A^{add} denote the partial score of the additional peaks of $\mathcal{S}^m$ under the null-model, i.e.*

$$A^{add} := \sum_{\substack{p^m \in \mathcal{S}^m \\ p^m \text{ additional}}} \Psi^{add}\left(p^m\right).$$

Then, the k-th moment of the partial score for additional peaks is

$$\mathbb{E}\left(\left(A^{add}\right)^k\right) \approx \sum_{p^m \in \mathcal{S}^m} \left(\Psi^{add}(p^m)\right)^k \cdot \prod_{m \in \mathcal{U}(p^m)} \left(1 - p^{\langle \ell \rangle}(m)\right).$$

Proof. A measured peak p^{m} is an additional peak if it cannot be matched to any predicted peak, i.e. there exists no predicted peak with mass inside the support of p^{m}:

$$\begin{aligned}\mathbb{E}(A^{\text{add}}) &= \mathbb{E}\left(\sum_{\substack{p^{\text{m}} \in \mathcal{S}^{\text{m}} \\ p^{\text{m}} \text{ additional}}} \Psi^{\text{add}}(p^{\text{m}})\right) \\ &= \mathbb{E}\left(\sum_{p^{\text{m}} \in \mathcal{S}^{\text{m}}} \Psi^{\text{add}}(p^{\text{m}}) \cdot \mathbb{1}_{\{\forall\, p^{\text{p}} \in \mathcal{S}^{\text{p}} : \mu^*(p^{\text{p}}) \notin \mathcal{U}^*(p^{\text{m}})\}}\right).\end{aligned}$$

Similar to the proof of Lemma 12.1, we can extract the summation and the score function from the expectation and switch to integer masses:

$$\mathbb{E}(A^{\text{add}}) \approx \sum_{p^{\text{m}} \in \mathcal{S}^{\text{m}}} \Psi^{\text{add}}(p^{\text{m}})\, \mathbb{E}\left(\mathbb{1}_{\{\forall\, p^{\text{p}} \in \mathcal{S}^{\text{p}} : \mu(p^{\text{p}}) \notin \mathcal{U}(p^{\text{m}})\}}\right).$$

The only difficulty is the expectation of the indicator function. Recall that this expectation is the probability of the corresponding event:

$$\mathbb{E}\left(\mathbb{1}_{\{\forall\, p^{\mathrm{p}} \in \mathcal{S}^{\mathrm{p}} : \mu(p^{\mathrm{p}}) \notin \mathcal{U}(p^{\mathrm{m}})\}}\right) \;=\; \mathbb{P}\left(\forall\, p^{\mathrm{p}} \in \mathcal{S}^{\mathrm{p}} :\; \mu(p^{\mathrm{p}}) \notin \mathcal{U}(p^{\mathrm{m}})\right).$$

The probability that a random predicted spectrum – induced by a random weighted string of length ℓ – has no peak of a particular mass m can be expressed in terms of the mass occurrence probabilities:

$$\mathbb{P}\left(\forall\, p^{\mathrm{p}} \in \mathcal{S}^{\mathrm{p}} :\; \mu(p^{\mathrm{p}}) \neq m\right) \;=\; 1 - p^{\langle \ell \rangle}(m).$$

More rigorously,

$$\mathbb{P}\left(\forall\, p^{\mathrm{p}} \in \mathcal{S}^{\mathrm{p}} :\; \mu(p^{\mathrm{p}}) \neq m\right) \;=\; \mathbb{P}\left(\mu(p^{\mathrm{p}}_1) \neq m, \ldots, \mu(p^{\mathrm{p}}_{N^{\langle \ell \rangle}}) \neq m\right),$$

where $N^{\langle \ell \rangle}$ is the number of peaks in a (randomly chosen) predicted peak list. It is the number of fragments in a random weighted string of length ℓ. If we additionally assume independence of mass occurrence probabilities (assumption (I)), this yields

$$\mathbb{P}\left(\forall\, p^{\mathrm{p}} \in \mathcal{S}^{\mathrm{p}} :\; \mu(p^{\mathrm{p}}) \notin \mathcal{U}(p^{\mathrm{m}})\right) \;=\; \prod_{m \in \mathcal{U}(p^{\mathrm{m}})} \left(1 - p^{\langle \ell \rangle}(m)\right),$$

which gives the stated result for $k = 1$. Similarly, we derive the higher moments. □

We are left with the moments of missing peaks.

Lemma 12.3 (Moments of missing peaks). *Under the conditions of Lemma 12.1, let A^{miss} denote the partial score of the missing peaks of a random peak list $\mathcal{S}^p$ under the null-model, i.e.*

$$A^{miss} \;:=\; \sum_{\substack{p^p \in \mathcal{S}^p \\ p^p \text{ missing}}} \Psi^{miss}\left(p^p\right).$$

Further, let $\mathcal{M}$ be the set of masses $\{1, 2, \ldots, m_{\max}\}$ considered in the experiment, and let $\mathcal{U} = \bigcup_{1 \le j \le |\mathcal{S}^m|} \mathcal{U}(p^m_j)$ be the set of all integer masses within the support of any measured peak. Then, the k-th moment of the partial score for missing peaks is

$$\mathbb{E}\left(\left(A^{miss}\right)^k\right) \;\approx\; \sum_{m \in \mathcal{M} \setminus \mathcal{U}} \left(\Psi^{miss}(m \cdot \Delta_m)\right)^k \cdot p^{\langle \ell \rangle}(m)$$

Proof. Since we assumed that predicted peaks do not have additional attributes, the missing score function $\Psi^{\mathrm{miss}}(\cdot)$ is a function of mass, i.e. $\Psi^{\mathrm{miss}}(p^{\mathrm{p}}) = \Psi^{\mathrm{miss}}(\mu^*(p^{\mathrm{p}}))$.

$$\mathbb{E}\left(A^{\mathrm{miss}}\right) = \mathbb{E}\left(\sum_{\substack{p^{\mathrm{p}} \in \mathcal{S}^{\mathrm{p}} \\ p^{\mathrm{p}} \text{ missing}}} \Psi^{\mathrm{miss}}\left(p^{\mathrm{p}}\right)\right),$$

and we can write the missing condition of the sum using an indicator function, switch to integer masses, and introduce a random peak P^{p}:

$$\mathbb{E}\left(A^{\mathrm{miss}}\right) \approx \mathbb{E}\left(\sum_{m\in\mathcal{M}\setminus\mathcal{U}} \Psi^{\mathrm{miss}}(m\cdot\Delta_m)\cdot\mathbb{1}_{\{\mu(P^{\mathrm{p}})=m\}}\right).$$

Again, the masses are non-random, thus

$$\mathbb{E}\left(A^{\mathrm{miss}}\right) \approx \sum_{m\in\mathcal{M}\setminus\mathcal{U}} \Psi^{\mathrm{miss}}(m\cdot\Delta_m)\cdot\mathbb{E}\left(\mathbb{1}_{\{\mu(P^{\mathrm{p}})=m\}}\right),$$

which we identify as the stated result for $k = 1$. Similarly, we derive the higher moments.

□

With the results of Lemmas 12.1– 12.3, the variances of the three partial scores A^{match}, A^{add} and A^{miss} are easily derived using the familiar identity $\mathrm{Var}(X) = \mathbb{E}(X^2) - (\mathbb{E}(X))^2$.

Lemma 12.4 (Variances of partial alignment scores). *Under the conditions of Lemma 12.1, the variances of the partial alignment scores are*

$$\begin{aligned}
\mathrm{Var}\left(A^{match}\right) &= \sum_{p^m\in\mathcal{S}^m}\sum_{m\in\mathcal{U}(p^m)}\left(\Psi^{match}(m\cdot\Delta_m,p^m)\right)^2\cdot p^{\langle\ell\rangle}(m) \;-\; \left(\mathbb{E}\left(A^{match}\right)\right)^2\\
\mathrm{Var}\left(A^{add}\right) &= \sum_{p^m\in\mathcal{S}^m}\left(\Psi^{add}(p^m)\right)^2\cdot\prod_{m\in\mathcal{U}(p^m)}\left(1-p^{\langle\ell\rangle}(m)\right) \;-\; \left(\mathbb{E}\left(A^{add}\right)\right)^2\\
\mathrm{Var}\left(A^{miss}\right) &= \sum_{m\in\mathcal{M}\setminus\mathcal{U}}\left(\Psi^{miss}(m\cdot\Delta_m)\right)^2\cdot p^{\langle\ell\rangle}(m) \;-\; \left(\mathbb{E}\left(A^{miss}\right)\right)^2.
\end{aligned}$$

Further,

$$\mathrm{Var}\left(A^{match}+A^{add}\right) = \mathrm{Var}\left(A^{match}\right) \;+\; \mathrm{Var}\left(A^{add}\right) \;-\; 2\cdot\mathbb{E}\left(A^{match}\right)\cdot\mathbb{E}\left(A^{add}\right).$$

Proof. The variances are obvious. The variance of the sum of the partial score of matching and additional peaks is computed via the relation

$$\mathrm{Var}(X+Y) \;=\; \mathrm{Var}(X) \;+\; \mathrm{Var}(Y) \;+\; 2\cdot\mathrm{Cov}(X,Y),$$

where $\mathrm{Cov}(A^{\mathrm{match}}, A^{\mathrm{add}}) = \mathbb{E}(A^{\mathrm{match}}\cdot A^{\mathrm{add}}) - \mathbb{E}(A^{\mathrm{match}})\cdot\mathbb{E}(A^{\mathrm{add}})$, and the expectation of the product is zero since each measured peak is either a matched or an additional peak. More precisely:

$$\mathbb{E}(A^{\mathrm{match}}\cdot A^{\mathrm{add}}) = \sum_{p^{\mathrm{m}}\in\mathcal{S}^{\mathrm{m}}}\mathbb{E}\left(X\cdot Y\right),$$

where X and Y refer to the matched and additional score of a certain measured peak p^{m}:

$$X := \sum_{m \in \mathcal{U}(p^{\mathrm{m}})} \sum_{p^{\mathrm{p}} \in \mathcal{S}^{\mathrm{p}}} \Psi^{\mathrm{match}}(m \cdot \Delta_m, p^{\mathrm{m}}) \cdot \mathbb{1}_{\{\mu(p^{\mathrm{p}})=m\}},$$
$$Y := \Psi^{\mathrm{add}}(p^{\mathrm{m}}) \cdot \mathbb{1}_{\{\forall p^{\mathrm{p}} \in \mathcal{S}^{\mathrm{p}}: \mu(p^{\mathrm{p}}) \notin \mathcal{U}(p^{\mathrm{m}})\}}.$$

We see that for all $p^{\mathrm{m}} \in \mathcal{S}^{\mathrm{m}}$, either X or Y is zero. Hence, the expectation is zero as is the sum of expectations. □

The moments of the alignment score distribution are the sum of the partial moments, regarding our independence assumption (I).

Theorem 12.5 (Moments of alignment scores). *Under the independence assumption (I), the expectation and variance of the optimal alignment score of a measured peak list $\mathcal{S}^m$ under a given scoring scheme score$(\cdot,\cdot)$ and an appropriate null-model on the predicted peak lists are*

$$\mathbb{E}\left(score\left(\mathcal{S}^p, \mathcal{S}^m\right)\right) = \mathbb{E}\left(A^{match}\right) + \mathbb{E}\left(A^{add}\right) + \mathbb{E}\left(A^{miss}\right),$$
$$\mathrm{Var}\left(score\left(\mathcal{S}^p, \mathcal{S}^m\right)\right) = \mathrm{Var}\left(A^{match} + A^{add}\right) + \mathrm{Var}\left(A^{miss}\right).$$

The alignment score expectation and variance can be computed in constant space and $\mathcal{O}(|\mathcal{M}|)$ time.

Proof. The equations are obvious by combining the previous Lemmas 12.1–12.4. For computing expectations and variances of partial matching and additional scores, we need to sum (or multiply) terms for each mass in a support of a measured peak. In total, these are $|\mathcal{U}|$ summations. Each such term can be computed in constant time. The expectation and variance for the missing peaks score involve summation over each mass in $\mathcal{M} \backslash \mathcal{U}$; each such term can again be computed in constant time. □

Note that we can compute the expected partial score of missing peaks $\mathbb{E}(X^{\langle \ell \rangle}) := \sum_{m \in \mathcal{M}} \Psi^{\mathrm{miss}}(m \cdot \Delta_m) \cdot p^{\langle \ell \rangle}(m)$ in a pre-processing step. We only need to do this for each string length ℓ of a database sequence. Similarly, we can compute $\mathrm{Var}(X^{\langle \ell \rangle})$ beforehand. These parameters correspond to the alignment score of an empty measured spectrum and a random sequence of length ℓ. Then, the missing score's moments can be computed for each measured spectrum and each string length in time $\mathcal{O}(|\mathcal{U}|)$ by subtracting the terms of masses contained in a support of a measured peak. Thus, for identification of a multitude of measured spectra and a comparably small number of different database sequence lengths, we can compute the alignment score parameters for k database sequences in time $\mathcal{O}(k \cdot |\mathcal{U}| + |\mathcal{M}|)$ compared to $\mathcal{O}(k \cdot |\mathcal{U}| + k \cdot |\mathcal{M}|)$ without this preprocessing step.

12.2. Computing p-values

Gaussian score distribution. Under the independence assumption (I), the alignment score distribution is the sum of independent random variables, and the Central Limit Theorem applies. To avoid confusion with previous notations, let

$$\tilde{\mu} := \mathbb{E}\left(score\left(\mathcal{S}^{\mathrm{p}}, \mathcal{S}^{\mathrm{m}}\right)\right),$$

and

$$\tilde{\sigma}^2 := \mathrm{Var}\left(score\left(\mathcal{S}^{\mathrm{p}}, \mathcal{S}^{\mathrm{m}}\right)\right)$$

denote the parameters as given in Theorem 12.5. Then

$$score\left(\mathcal{S}^{\mathrm{p}}, \mathcal{S}^{\mathrm{m}}\right) \sim \mathrm{Norm}\left(\tilde{\mu}, \tilde{\sigma}^2\right).$$

Suppose we are given a measured peak list $\mathcal{S}^{\mathrm{m}}$ and a scoring scheme $score(\cdot,\cdot)$. Suppose further that $a := score(\mathcal{S}^{\mathrm{p}}(s^{\langle\ell\rangle}), \mathcal{S}^{\mathrm{m}})$ for a certain peak list predicted from a database sequence $s^{\langle\ell\rangle}$ of length ℓ. Then, a statistical significance of the alignment score can be computed by the one-sided p-value for the right tail,

$$\mathbb{P}\left(score(\mathcal{S}^{\mathrm{p}}, \mathcal{S}^{\mathrm{m}}) \geq a\right) \approx \mathbb{P}\left(Z \geq \frac{a - \tilde{\mu}}{\tilde{\sigma}}\right),$$

where $\mathcal{S}^{\mathrm{p}}$ is a random predicted spectrum under an appropriate null-model, $Z \sim \mathrm{Norm}(0,1)$ a standard Gaussian random variable, and $\tilde{\mu}, \tilde{\sigma}^2$ are the computed expectation and computed variance of the peak list alignment score under the appropriate null-model. A similar approach was also taken in [8].

Non-Gaussian score distribution. If for any reason the alignment score distribution cannot be approximated adequately by a Gaussian distribution, we may use general inequalities for sums of independent random variables to give lower bounds for the p-value of a computed alignment score.

Let us again assume that peak masses occur independently and that we can treat the alignment score distribution as a sum of independent random variables. Using Lemmas 12.1–12.3, we can compute arbitrary moments of the partial alignment score distributions, given these moments exist. The k-th moment of the alignment score distribution is given by

$$\mathbb{E}\left(A^k\right) \;=\; \mathbb{E}\left(score\left(\mathcal{S}^{\mathrm{p}}, \mathcal{S}^{\mathrm{m}}\right)^k\right) \;=\; \mathbb{E}\left(\left(A^{\mathrm{match}} + A^{\mathrm{add}} + A^{\mathrm{miss}}\right)^k\right).$$

Assuming independence of the three partial scores and given their moments up to order k, we can therefore compute the k-th moment of the alignment score distribution by computing the polynomial $\left(A^{\mathrm{match}} + A^{\mathrm{add}} + A^{\mathrm{miss}}\right)^k$ of degree k. Noting that $\mathbb{E}(X^q Y^r) = \mathbb{E}(X^q)\,\mathbb{E}(Y^r)$ for independent random variables X and Y and any $q, r \geq 1$,

and that $\mathbb{E}(X+Y) = \mathbb{E}(X) + \mathbb{E}(Y)$ for any two random variables X and Y, each term of this polynomial can be computed using the moments of the partial score.

For computing the p-value of a certain alignment score a, we can use classical inequalities to give estimates on the deviation from the mean that are valid for any alignment score distribution under independence assumptions. Naturally, since these inequalities hold for any distribution, they usually yield much more conservative p-values.

Lemma 12.6 (Lower bounds for p-values). *Let X_i, $i = 1 \dots n$, be independent real random variables and let $S = \sum_{i=1}^{n} X_i$ be their sum. Let further $\mu_k = \mathbb{E}(S^k)$ be the k-th moment of S, let σ be the standard deviation of S and let $\alpha_k = \mu_k/\sigma^k$. Then, lower bounds for p-values of the centered random variable $S - \mathbb{E}(S)$ are provided by the following three inequalities:*

The general Chebychev inequality,

$$\mathbb{P}(S - \mathbb{E}(S) \geq t\sigma) \leq \frac{1}{1+t^2}.$$

If additionally each summand X_i is bounded s.t. $\mathbb{P}(X_i \in [a_i, b_i]) = 1$, the Hoeffding inequality applies:

$$\mathbb{P}(S - \mathbb{E}(S) \geq nt) \leq \exp\left(-\frac{2n^2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right).$$

For moments up to order 4, we may use the Zelen inequality:

$$\mathbb{P}(S - \mathbb{E}(S) \geq t\sigma) \leq \frac{1}{1 + t^2 + \frac{(t^2 - t\alpha_3 - 1)^2}{\alpha_4 - \alpha_3^2 - 1}}.$$

Proof. See [16,17] for proofs. □

Chebycheff's inequality uses the first two moments, it was also used in [8] for p-value estimation. Hoeffding's inequality also uses the first two moments, whereas Zelen's inequality uses the first four moments and thus also considers the skew and kurtosis of the distribution.

12.3. p-value Scores

Since the peak list alignment score is an additive score, its value and distribution depends on the number of peaks in the measured and the predicted spectrum. This makes it difficult to compare alignment scores for different measured spectra and sequence lengths.

We may circumvent this problem, if we do not use the alignment score for ranking candidate sequences, but its p-value score.

Definition 12.7 (p-value score). Suppose the alignment score of a predicted peak list $\mathcal{S}^{\mathrm{p}}$ and a measured peak list $\mathcal{S}^{\mathrm{m}}$ has significance $\tilde{p}$. Then the *p-value score* of the alignment is

$$score_{p\text{-val}}(\mathcal{S}^{\mathrm{p}}, \mathcal{S}^{\mathrm{m}}) := -\log_{10}(\tilde{p}).$$

For each pair of measured and predicted peak lists, the alignment score is computed as described in Chapter 11, its distribution is estimated using the method described described above, and finally the p-value $\tilde{p}$ of the alignment score is computed. We then use the p-value score for ranking the candidate sequences. This method is applicable for *any* underlying scoring scheme; a similar method was shown to be effective for PFF data [125].

p-value scores also provide a possibility to compensate the different null-models, introduced by the different sequence lengths (cf. Chapter 12), at least partially.

12.4. Numerical Evaluation

The above calculations are only valid under the independence assumption (I), although the random variables are slightly correlated. Moreover, a sufficiently large number of summands is needed to apply the Central Limit Theorem. To show that our estimations are reasonable in application settings, we have performed numerical simulations with a mass precision of $\Delta_m = 0.1$ Da.

Estimation of score distribution. We use the following simple scoring scheme: Peaks are matched with score 1, if their mass difference is lower or equal 1 Da. Additional and missing peaks are penalized by -0.2. This is almost the peak-counting score of Example 11.6, except we also penalize additional and missing peaks.

We used a measured peak list from a proteomics experiment on *Corynebacterium glutamicum* with 22 peaks in the mass range 500–3 500 Da. The peaks had the following masses (in Da, truncated at two decimals) and absolute intensities (truncated at two decimals) as detected by the manufacturer's processing software:

Peak no.	1	2	3	4	5	6
Mass (Da)	917.53	1 046.50	1 086.54	1 113.65	1 310.25	1 347.69
Abs. int.	644.88	20 052.81	822.64	951.77	1 653.31	7 677.34
Peak no.	7	8	9	10	11	12
Mass (Da)	1 363.69	1 549.76	1 779.90	1 800.89	1 833.90	1 862.89
Abs. int.	4 222.92	1 454.08	1 014.07	5 113.29	2 500.23	9 922.84
Peak no.	13	14	15	16	17	18
Mass (Da)	1 990.69	2 093.04	2 104.51	2 119.07	2 124.08	2 141.06
Abs. int.	423.46	1 306.04	529.33	5 532.20	10 990.90	663.20
Peak no.	19	20	21	22		
Mass (Da)	2 370.21	2 469.26	2 678.46	2 778.38		
Abs. int.	355.20	8 719.27	620.60	909.80		

We aligned this peak list with 25 000 peak lists predicted from i.i.d. random sequences of length $\ell = 500$, $\ell = 1\,000$, and $\ell = 4\,000$, using Swiss-Prot amino acid frequencies for the sequence sampling. The empirical density of the alignment scores was estimated for each length and compared with a Gaussian distribution with parameters computed as described above. The empirical and computed moments agree reasonably well, as the following table shows for the expectation, variance, and the standard deviation:

	$\ell = 500$		$\ell = 1\,000$		$\ell = 4\,000$	
	Empirical	Computed	Empirical	Computed	Empirical	Computed
$\mathbb{E}$	-10.06	-10.28	-15.67	-16.07	-49.51	-50.19
Var	1.50	1.97	2.93	3.85	10.60	13.96
$\sqrt{\text{Var}}$	1.22	1.40	1.71	1.96	3.25	3.74

Let again A denote the alignment score of the measured peak list and a random predicted peak list. Then

$$A_{\text{norm}} := \frac{A - \mathbb{E}(A)}{\sqrt{\text{Var}(A)}}$$

is the normalized score of zero mean and unit variance. Under our independence assumption (I), this normalized alignment score thus has a standard Gaussian distribution:

$$A_{\text{norm}} \sim \text{Norm}(0, 1).$$

This distribution is the same for all sequence lengths, measured spectra and scoring schemes.

In Figure 12.2, the empirical distributions of the normalized scores are compared to standard Gaussian distributions of zero expectation and unit variance. On the left panel, the density functions are compared. They show a reasonable agreement for all protein lengths $\ell = 500$ (top), $\ell = 1\,000$ (center), and $\ell = 4\,000$ (bottom). Note that both empirical density functions are skewed and are thus not symmetric. In fact, their right tail seems to be heavier than the left tail. The right panel gives quantile-quantile plots of the empirical distributions and standard Gaussian distributions together with the bisectors (lines). Points on the line would indicate a perfect agreement of the two distributions. This is clearly not the case here. The center and the right tail of the distributions are in good agreement, whereas the agreement of the left tails is not as good. However, the agreement gets better for increasing protein length. Moreover, we compute the probability that the score exceeds some given value, not the probability that it deviates from a given value. Thus, we are computing one-sided p-values on the right tails, which are well approximated by the standard Gaussian.

Discrete score distributions. Besides the obvious reason that we do not want to simply ignore additional and missing peaks, the introduction of penalties in the scoring scheme also has the desirable side-effect that the alignment score distribution is smoothed even for moderate numbers of measured peaks. Without this smoothing, a much larger number of measured peaks is necessary to gain a Gaussian-like score distribution. To demonstrate this, we use the peak-counting score of Example 11.6 with a threshold of 1 Da

for the mass difference and score 0 for both additional and missing peaks. We again draw 25 000 random spectra and align them with the same measured peak list as before. As we can see in Figure 12.3 (left), the normalized score is no longer unimodal: It can only take discrete values $0, 1, 2, \ldots$. The approximation by a standard Gaussian distribution is clearly not valid. Using a different peak-detection algorithm, we generate a second peak list for the same measured raw spectrum. This detection algorithm uses a much lower threshold, resulting in 373 peaks in the peak list. Then, although the score distribution is still discrete and restricted to integers, the estimated density function is much smoother and the approximation by a standard Gaussian seems more accurate. The normalized density function and the standard Gaussian density are given in Figure 12.3 (right). Note, however, that the estimated mean and variance are still within reasonable error bounds for both alignment score distributions. For the smaller peak list, we compute an expected alignment score of 0.97 ± 0.96 and an empirical expectation of 0.88 ± 0.91. For the larger peak list, the expected alignment score is 16.95 ± 3.99 and its empirical counterpart is 15.86 ± 3.93. This allows us to still use the inequalities of Section 12.2 for estimating the p-value for general score distributions.

Moments as function of sequence length. In Section 9.2, we saw that the fragment mass occurrence probabilities depend smoothly on the protein length. Since the moments of the alignment score distribution are computed from these occurrence probabilities, we can expect that the moments also depend smoothly on the protein length. To validate this, we use the peak-counting score with penalties -0.2 as described above and compute the expectation and variance of the alignment score for the smaller peak list given above for protein lengths from $\ell = 500$ up to $\ell = 10\,000$ in steps of 500. Further, we randomly generate 10 000 predicted spectra for each sequence length and estimate the empirical parameters of the alignment score. The expected alignment score seems to depend linearly on the protein length, see Figure 12.4 (left). Moreover, the empirical data is in good agreement with the model for all lengths. Not surprisingly, the dependence of the standard deviation on the protein length is non-linear but smooth nonetheless. The agreement of empirical data and the model is also reasonable, although it deviates with increasing protein length. Our model seems to overestimate the standard deviation, which results in an underestimation of the statistical significance. Thus, the estimation error is towards the safe side: A broader distribution with the same expectation than a narrower one results in a more conservative p-value.

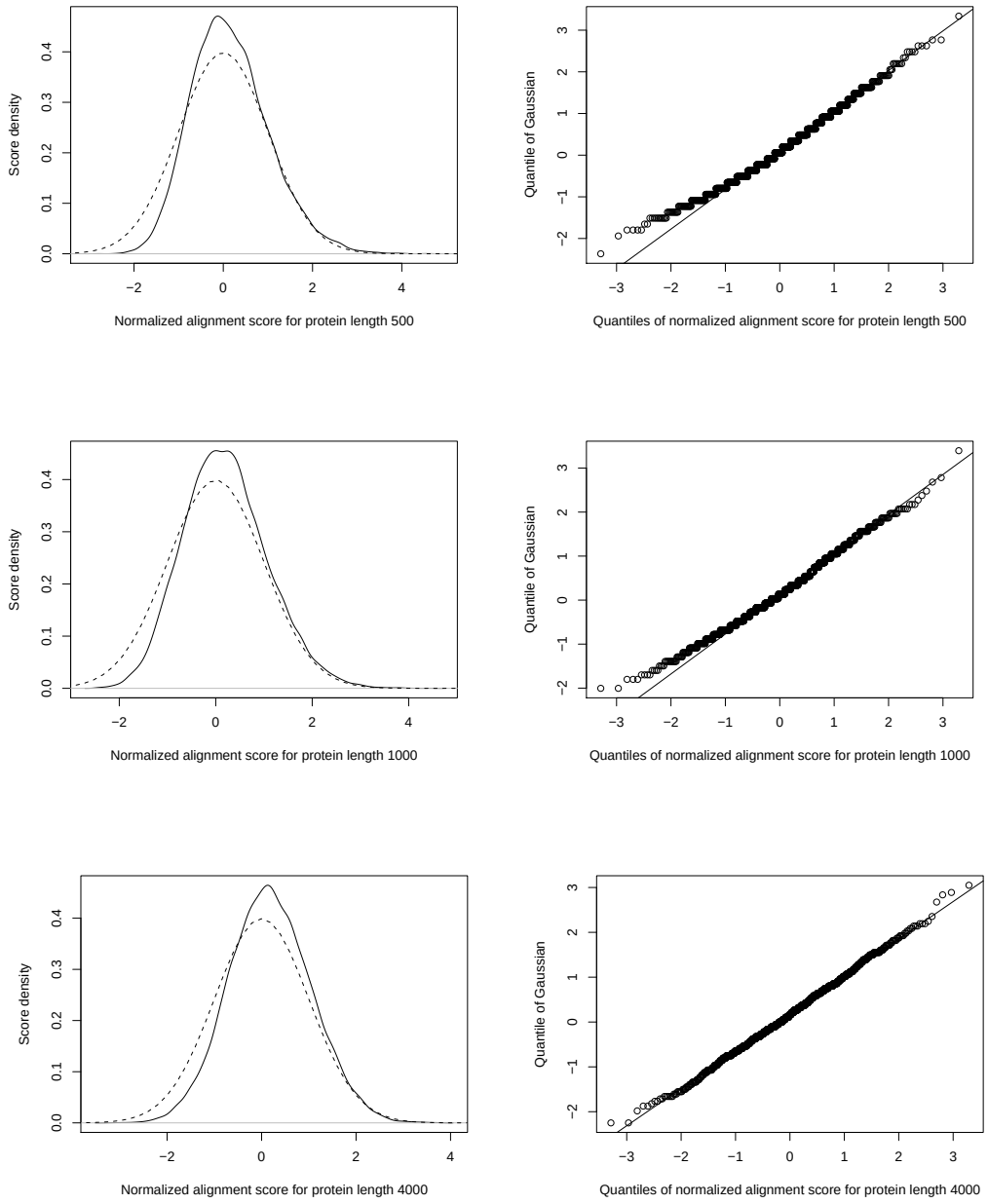

Figure 12.2.: Estimated normalized alignment score distributions for protein sequences of length $\ell = 500$ (top), $\ell = 1\,000$ (center), and $\ell = 4\,000$ (bottom). Left panels: Density plots. Solid line: Empirical scores of 25 000 simulated sequences. Dashed line: Standard Gaussian density. Right panels: Quantile-quantile plots vs. standard Gaussian distribution, solid lines denote bisectors.

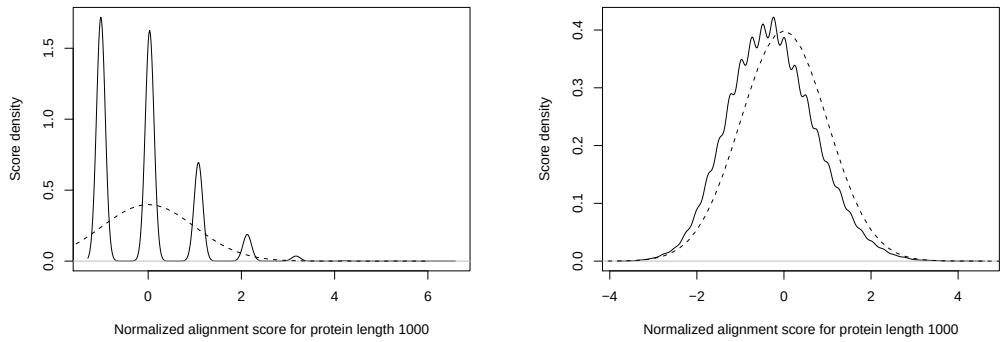

Figure 12.3.: Normalized alignment score distribution density for peak-counting score and 25 000 simulated random spectra. Solid line: Empirical score distribution. Dashed line: Standard Gaussian density. Left: Measured peak list with 22 peaks. Right: Measured peak list with 373 peaks generated from same raw data.

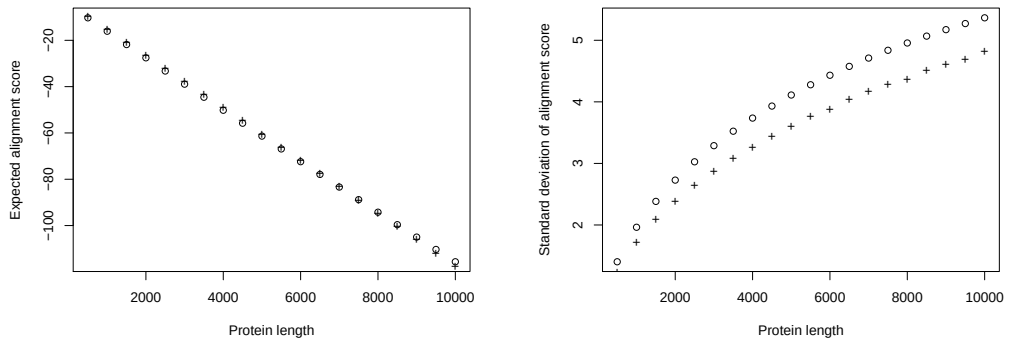

Figure 12.4.: Expectation and standard deviation of alignment scores for given measured peak list as function of protein length. Bullets: Computed moments. Pluses: Empirical moments estimated from 10 000 sequences each. Left: Expectation. Right: Standard deviation.

13. Evaluation

So far, we were concerned with theoretical aspects of peak list alignments. We only considered some artificial examples involving derivates of the simple peak-counting score introduced in Example 11.6. The goal of this chapter is two-fold: To develop practical peak scoring schemes and to prove the applicability of the approach on real data. We discuss several aspects of practical scoring schemes including the use of intensity information in the peak scoring and provide simulated data to demonstrate accuracy of the score parameter estimation. In particular, we develop the family of Gaussian scoring schemes and apply some of its members in an identification experiment using real proteomics data to prove the applicability of our framework on these data. We compare our results with the standard software MASCOT.

13.1. Scoring Schemes

Although a measured peak is described at least by its mass and intensity, most identification algorithms only use its mass [54]. This is partly because mass is the most discriminative parameter measured and partly because the peak's intensity crucially depends on the actual parameter settings of the machine. The basis for many schemes is the observation that a measurement error between the "real" mass of a molecule and the measured mass can be described by a Gaussian distribution with zero mean and a standard deviation *sd* dependent e.g. on the machine settings. In practice, the mean might also deviate from zero if the machine is not calibrated correctly. We will not consider this problem in our scoring schemes; it should be addressed by a more sufficient calibration or a re-calibration of the peak lists.

The incorporation of additional attributes like peak intensities may be of particular importance when scoring missing and additional peaks. For missing peaks, recall that we have transformed the raw data of the mass spectrum into a peak list discarding candidates whose intensity falls below a given threshold. Hence, slight changes of this threshold can dramatically change scores that do not take peak intensities into account. For additional peaks, similar arguments apply.

As an example for practical scoring schemes, we develop the family of *Gaussian scoring schemes*. It uses ideas shown to be useful in other identification algorithms for both PMF and tandem MS data and allows flexible integration of peak intensity data.

Mass difference. The matching score for two peaks p^{p} and p^{m} should reflect the Gaussian mass error distribution. Thus, it should decrease exponentially with increasing mass difference. In the Gaussian scoring scheme family, we compute the probability that a

zero mean Gaussian random variable deviates from zero by at least the difference of the peak masses. The standard deviation *sd* of the distribution is taken as a user-defined parameter; it might be set to model the accuracy of the MS instrument.

This matching score does not provide a finite support for all measured peaks. We therefore use a threshold of 0.05 and set the score to $-\infty$ if it would drop below this threshold. This threshold corresponds to a mass difference of about $2\,sd$. Let $Z \sim \mathrm{Norm}(0, sd)$ be a Gaussian random variable, then the matching score function is

$$\Psi^{\mathrm{match}}\left(p^{\mathrm{p}}, p^{\mathrm{m}}\right) \;=\; \mathbb{P}\left(|Z| \geq \left|\mu^*(p^{\mathrm{p}}) - \mu^*(p^{\mathrm{m}})\right|\right)$$

whenever this score is above 0.05.

A similar approach is taken in ProFound [139] and the tandem MS identification software SCOPE [8], whereas MASCOT uses a constant positive matching score similar to that of Example 11.6.

Note that this matching score function is just one example. If for any reason we would like to model another mass error distribution, we just have to define another matching score function.

Robust incorporation of intensities. In order to incorporate intensities of measured peaks into the scoring scheme, we rank all peaks in the peak list in ascending order according to their absolute intensity. Then, the intensity of the first 10% of the ranked peaks is set to 0, whereas the intensity of the last 10% of the ranked peaks is set to 1. The intensities of the remaining peaks are scaled linearly between 0 and 1 according to their respective absolute intensity. Thus, a chemical noise peak with high intensity or a small number of incorrectly detected peaks with very low intensity cannot spoil the identification of the whole peak list. We use intensities only for measured peaks. Given an appropriate prediction model as proposed e.g. in [54, 118], it would also be possible to incorporate intensities of predicted peaks.

Let $0 \leq \mathrm{int}(p^{\mathrm{m}}) \leq 1$ denote a measured, re-scaled intensity value of a peak, computed by the procedure described above. We compute an intensity scaling factor

$$f \;:=\; \frac{1 + 2 \cdot \mathrm{int}(p^{\mathrm{m}})}{3}$$

and use it to scale the previously computed matching score for mass differences, resulting in the modified matching score function

$$\Psi^{\mathrm{match}}\left(p^{\mathrm{p}}, p^{\mathrm{m}}\right) \;=\; f \cdot \mathbb{P}\left(|Z| \geq \left|\mu^*(p^{\mathrm{p}}) - \mu^*(p^{\mathrm{m}})\right|\right).$$

Again, we set the value of this function to $-\infty$ if the mass difference exceeds a certain threshold. The matching score function for mass differences is thus multiplied by 1 for the peaks of highest intensity and reduced to 1/3 of its value for low intensity peaks. The approach is suitable for any other transformation of intensity information, such as logarithmic transforms proposed e.g. in [132].

Scoring gap peaks. For the family of Gaussian scoring schemes, we also want to use the measured peak's scaled intensity for scoring additional peaks. In particular, we want to score peaks of high intensity with a higher penalty than low intensity peaks. Thus, a simple additional scoring function is

$$\Psi^{\text{add}}(p^{\text{m}}) = -c^{\text{add}} \cdot \text{int}(p^{\text{m}})$$

for a user-defined constant $c^{\text{add}} \geq 0$. Additional peaks with very low intensity are then penalized by 0 and thus simply ignored, and additional peaks of high intensity are highly penalized. It would also be possible to use the fragment mass distributions of Section 8.4 for adjusting the penalty by the probability that the observed mass is the mass of a peptide and not a contaminant. We will not explore this further.

Since we do not have intensity information for predicted peaks, missing peaks are always penalized with a constant penalty $c^{\text{miss}} \geq 0$:

$$\Psi^{\text{miss}}(p^{\text{p}}) = -c^{\text{miss}}.$$

Note that, once peak intensities can be predicted from the peak mass, it is easy to incorporate this knowledge in the missing score function.

13.2. Evaluation on Proteomics Data

To evaluate our method, 325 PMF tryptic mass fingerprints of charge state $(M+H)^+$ from an in-house proteomics experiment on the organism *Corynebacterium glutamicum* (Cg) were measured on a Bruker Ultraflex mass spectrometer. The proteins were separated using SDS-PAGE before mass measurement, so Carbamidomethyl was set as a fixed mass modification of $\approx +57$ Da for Cysteine. Further, two different peak lists of different sizes were extracted from each measured raw spectrum. The identification was then run on a Cg protein sequence database and a modified version of the Swiss-Prot database.

Processing the raw spectra. To assess robustness and flexibility of the method, two different peak lists were derived for each raw spectrum. The first peak list was taken from the manufacturer's peak detection software: This software is conservative in picking only peaks with high intensity. The resulting peak lists were comparatively small. They contained up to 90 peaks with an average of 20 peaks. We will refer to these peak lists as "Bruker" or manufacturer's peak lists. The second peak list was taken from a peak detection algorithm developed in our group. This algorithm computed much larger peak lists of 34 to 729 peaks with an average of 277 peaks. A comparison of the distribution of the number of peaks in the two peak lists is shown in Figure 13.1. For unknown reasons, the manufacturer's software delivered only 316 non-empty peak lists, whereas in 9 of the raw spectra, no peaks were detected. The other algorithm delivered 325 valid peak lists. For better comparison, we differentiate the peak lists delivered by the algorithm of our group in the following by "PL" and "PL_{316}", denoting the whole set of

325 peak lists and the set of the 316 peak lists with corresponding non-empty peak list from the manufacturer's peak detection. Due to the different peak detection, the valid mass ranges for the measured and predicted peak lists were set to 500–3 000 Da for the manufacturer's software, and to 800–3 000 Da for the PL peak lists. All peaks outside this range were discarded.

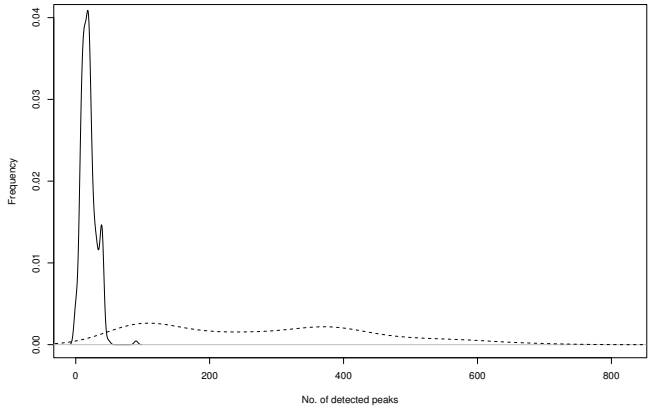

Figure 13.1.: Density estimates of peak list sizes. Solid line: Peak-list sizes from Bruker peak detection software. Dashed line: Size of peak lists from an in-house peak detection software.

Identification procedure. The identification procedure is as follows: In a first step, both sets of peak lists are identified using an in-house Cg protein sequence database derived from experimental expressed sequence tags (EST) data. This database contains 3 510 organism-specific sequences. Since the "true" identification of the mass spectra is unknown, we only record the sequence identifier of the best scoring protein sequence for each peak list. We repeat this procedure for both MASCOT versions and our framework with several parameter sets. In a second step, we merge the Cg sequence database with a modified version of the Swiss-Prot sequence database with 155 824 sequence entries, resulting in a total of 159 334 sequences. The Swiss-Prot database is modified to avoid getting protein sequences from other species that are very similar to the "correct" corresponding Cg sequence. This is achieved by excluding sequences from the Swiss-Prot database that are too similar to any Cg sequences. The threshold for exclusion is a BLASTp [4, 5] e-value of 10^{-30} or better, leading to an exclusion of 38 493 sequences. This new database can be interpreted as a very noisy version of the original Cg database. The paper [65] also describes several other techniques for generating noisy databases. Similar to the method used in [79], a peak list identification is assumed to be correct whenever the identified sequence belongs to the Cg database (i.e. had an identifier from the Cg database) and is the same as in the Cg database run.

Both sets of peak lists are identified using the framework of Chapters 11 and 12 with

the p-value score of Section 12.3, based on a Gaussian scoring scheme as described in Section 13.1. Intensities are scaled as described in Section 13.1, where the 10% highest and lowest values were set to 1 and 0, respectively.

For later use, we introduce the following two parameter sets, each with a particular standard deviation, penalties for additional and missing scores and use of intensities as described above. We denote them by (A) and (B):

Parameter	std. dev. sd	add. penalty c^{add}	miss. penalty c^{miss}	intensity used
A	0.8	-0.1	-0.1	No
B	0.8	-0.3	-0.4	Yes

To validate the use of a Gaussian distribution for approximating the alignment score distribution and thus also to validate the derived p-value score, we performed numerical simulations by aligning 25 000 random sequences of length 2 000 to the same measured peak list of 373 peaks as in Section 12.4. We again observe a good agreement between the empirical score distribution and the estimated Gaussian, see Figure 13.2 for quantile-quantile plots of normalized alignment scores using the Gaussian scoring scheme with both parameter sets (A) and (B).

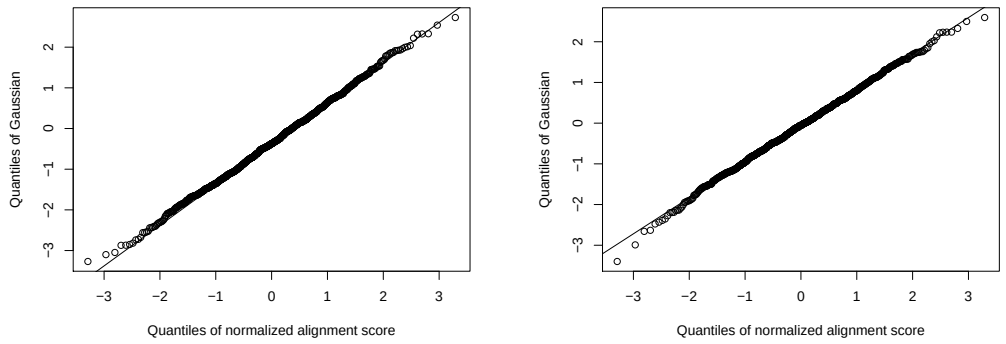

Figure 13.2.: Quantile-quantile plots for normalized alignment scores of 25 000 simulated protein sequences of length 2 000 and a measured peak list for the Gaussian scoring scheme. Left: Parameter set (A). Right: Parameter set (B).

We compare the identification results of our framework to those of both MASCOT versions v1.9 and v2.1. A maximal mass difference of 1 Da is used for MASCOT.

Results. Not surprisingly, different parameter sets lead to different numbers of correct identifications (cf. Table 13.1). Nevertheless, these numbers do not change rapidly with changing parameters, indicating a robust behavior of the alignment identification procedure. Using the manufacturer's peak lists, a small penalty of additional and missing

Method		w/out intensity			w/ intensity		
		Bruk.	PL	PL_{316}	Bruk.	PL	PL_{316}
MASCOT v1.9		123	58	53	-	-	-
MASCOT v2.1		119	59	53	-	-	-
SAMPI							
c^{add}	c^{miss}						
-0.1	-0.1	112	56	51	72	106	87
-0.2	-0.2	111	56	51	78	96	92
-0.3	-0.3	96	54	48	65	103	98
-0.3	-0.4	89	53	48	52	110	105
-0.4	-0.4	91	54	49	54	108	103
-0.5	-0.4	94	53	48	57	109	104

Table 13.1.: Number of correctly identified spectra. Method: MASCOT or SAMPI. Different parameter sets for the latter. All parameter sets tested with and without use of peak intensities. There are 316 peak lists in the Bruker (Bruk.) and PL_{316} peak list sets and 325 peak lists in the PL peak list set.

peaks yields a comparable number of correct identifications as MASCOT. Using peak intensities in the scoring scheme, this number drops considerably. An explanation for this phenomenon is that these peak lists already consist of the highest abundant peaks, which are now scaled from 1/3 to 1, distorting the relevance of peaks. Using the larger, noisier peak lists results in the completely opposite behavior: Now, without using intensities to discriminate important and non-important peaks, the identification rate drops to about 1/2 for both the Gaussian-p-value schemes and MASCOT. Using peak intensities in addition leads to a good identification rate again. Note that now higher penalties for additional and missing peaks are also helpful.

In all cases, we found the score separation of correct and incorrect identifications to be comparable to MASCOT. In Figure 13.3, we give receiver operating characteristics (ROC) plots for MASCOT and the Gaussian-p-value scores with parameter sets (A) and (B). ROC plots are a visual tool that give the false positive rate compared to the true positive rate. A perfect identification algorithm would have zero false positives for any number of true positives, i.e., it would always identify correctly. This would result in a horizontal line at ordinate 1. A bisector would correspond to an identification algorithm that uniformly selects a true or false answer at random. ROC plots are a standard tool for comparing sequence analysis algorithms [57]; they are also used for comparing protein identification algorithms [123].

For the Bruker peak lists, the plots indicate a score separation inferior to that of MASCOT for both parameter sets (A) and (B). However, the number of correctly identified proteins is only slightly smaller for parameter set (A) (cf. Table 13.1). For the PL_{316} peak lists, MASCOT shows a considerably worse score separation. This is mostly due to the fact that MASCOT does not use peak intensity information. Both Gaussian scoring schemes show a comparable behavior. They both separate true and false positives bet-

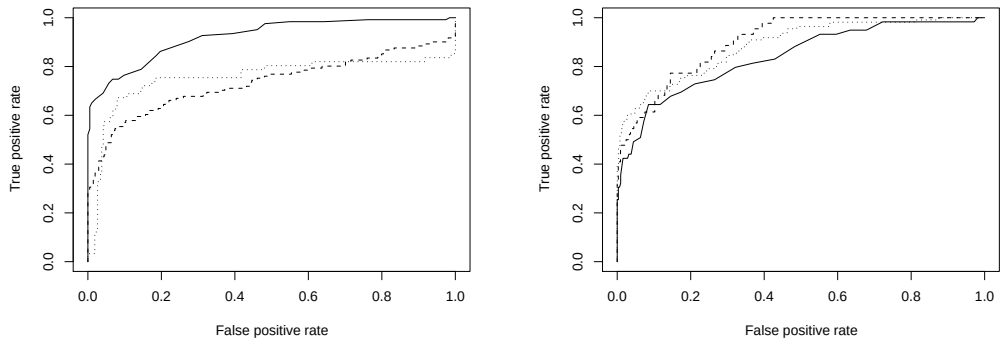

Figure 13.3.: Receiver operating characteristics (ROC) curves for identification. Solid line: MASCOT v2.1. Dashed line: Gaussian-p-value score with parameter set (A). Dotted line: Same with parameter set (B). Left: Bruker peak lists. Right: PL_{316} peak lists.

ter than MASCOT, but only parameter set (B) uses intensity information to distinguish between important and non-important peaks. This is reflected in the almost two-fold number of correct identifications compared to parameter set (A) and MASCOT. Note that additionally using intensities in parameter set (A) also increases the number of correct identifications to a similar level.

In general, the identification rate of our method is about 90% of MASCOT's identification rate on Bruker peak lists. Using the trimmed relative intensities in the scoring distorts the importance of peaks in small peak lists containing only peaks of high intensities. This problem might be circumvented when using rank statistics on the intensities instead. For noisy peak lists, incorporating intensity information into the scoring is crucial to distinguish important and non-important peaks. The separation of true and false positive scores of our method indicates some problems with the p-value score. Since we compute and compare p-values under different null-models for different sequence lengths, the p-values might still depend on the sequence lengths, causing the weak score separation. This problem may be solved by normalizing the score for measured peak list size and sequence length. However, there seems to be no obvious normalizing score transformation.

14. Conclusion

Several aspects of mathematical problems arising in the identification of peptide mass fingerprints using sequence databases have been discussed.

We have presented a model of random weighted strings together with cleavage schemes as a model for cleavage fragments of random protein sequences.

For computing statistics of cleavage fragments, we have introduced the general framework of weighted hidden Markov models and Markov additive chains. In particular, we have investigated the distribution of length, mass and number of fragments as well as fragment mass occurrence probabilities in finite random protein sequences. We have also developed recurrence equations for these statistics as an additional computational tool, and provided efficient dynamic programming algorithms. In contrast to many existing approaches, our method relies on a sequence database only for estimating character frequencies; fragment statistics are independent of database size and composition.

We have presented SAMPI: A general protein identification framework based on peak list alignments and peak-wise scoring schemes. SAMPI allows consistent handling of mass accuracies, additional peak attributes such as peak intensities, as well as additional and missing peaks. We have provided a family of scoring schemes that uses several previously published ideas.

Based on fragment statistics of the first part, we have given a general method for estimating the p-value of an alignment score. The estimation procedure is deterministic and independent of the size of the sequence database; it does not use time-consuming sampling. We have introduced p-value scores to reduce the influence of peak list size and sequence length on the score and allow direct comparison of peak list alignment scores for different measured and predicted peak list sizes.

Finally, we have demonstrated the applicability of our alignment and significance computation frameworks on real proteomics data, and have compared our results to the results of the standard software MASCOT.

Open Problems

Several problems in both fragment statistics and peak list alignment remain unsolved or have been opened by the thesis.

We have only considered the i.i.d. random sequence model for developing and computing fragment statistics. It would be interesting to also take into account dependencies among amino acids in the protein sequence by extending the statistics to Markov random sequence models. One evident advantage of the most simple Markov model of order one would be the possibility to explicitly model the methionine prefix found in the first

fragment. Note that our random weighted string model, the weighted HMMs, and the additive Markov chains are defined for any random sequence model. Nevertheless, most recurrence equations and the efficient algorithms are only valid for the i.i.d. case.

The mass distributions of fragments are periodic in the sense that the single "curves" were separated with period 20. We have given some hints from the theory of linear Diophantine equations, but have not studied these hints further. This might be an interesting issue for further investigation.

We already gave one example of a protease that is not covered by our cleavage scheme model. One open problem is the extension of the weighted HMM model to more complex cleavage patterns involving more than two characters. This would also allow the investigation of DNA fragmentation by RNAses. Further, it might be necessary to allow pairs of cleavage/prohibition characters in a cleavage scheme such that not every prohibition character suppresses every cleavage character.

As for the peak list alignment framework for protein identification, we did not discuss the problem of incomplete cleavage, where a potential cleavage is not performed in all copies of the protein, leading to more fragment peaks in the spectrum. These can easily be taken care of when computing the predicted spectrum from a database sequence, but the fragment statistics do not capture these fragment masses. Including missed cleavage sites in the fragment statistics is straightforward, but fragment masses may now become depended; for a mass of a fragment including one missed cleavage site, there should be two fragment masses in the peak list that sum up to this mass.

The ROC curves show a weak separation of scores of true and false positives. This might indicate a problem with the comparability of scores. It is an open problem whether the alignment or p-value scores can be normalized by peak list size and sequence length for improving the score separation.

As a more theoretical aspect, we did not provide any constraints on the scoring schemes that guarantee a Gaussian alignment score distribution important for the significance computation. It would be interesting to see if such constraints can be found, and if classes of practical scoring schemes can be described that fulfill these constraints.

Further, tandem mass spectrometry has gained a lot attention in the last years. Protein identification by sequence database searching and tandem mass spectra has a lot of problems common to protein identification by peptide mass fingerprinting. However, the observed fragment masses are highly dependent in tandem mass spectrometry. It is an open question whether corresponding fragment statistics based on random weighted strings can be developed for tandem mass spectrometry. On the algorithmic side, peak list alignments can be used for any type of mass spectrometry, given adapted scoring schemes. Many scoring methods can be found in the literature and it would be interesting to see whether some of them can be re-implemented as alignment scoring schemes.

Bibliography

[1] R. Aebersold. A mass spectrometric journey into protein and proteome research. *J. Am. Soc. Mass Spectrom.*, 14:685–695, 2003.

[2] R. Aebersold and M. Mann. Mass spectrometry-based proteomics. *Nature*, 422:198–207, 2003.

[3] T. Aittokallio, P. Ojala, T. J. Nevalainen, and O. Nevalainen. Automated detection of differently expressed fragments in mRNA differential display. *Electrophoresis*, 22(10):1935–1945, 2001.

[4] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J. Mol. Biol.*, 215:403–410, 1990.

[5] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, 25:3389–3402, 1997.

[6] R. J. Arnold and J. P. Reilly. Fingerprint matching of E. coli strains with matrix-assisted laser desorption/ionization time-of-flight mass spectrometry of whole cells using a modified correlation approach. *Rapid Commun. Mass Spectrom.*, 12:630–636, 1998.

[7] A. E. Ashcroft. Protein and peptide identification: the role of mass spectrometry in proteomics. *Nat. Prod. Rep.*, 20:202–215, 2003.

[8] V. Bafna and N. Edwards. SCOPE : A probabilistic model for scoring tandem mass spectra against a peptide database. *Bioinf.*, 17:S13–S21, 2001.

[9] V. Bafna and N. Edwards. On de novo interpretation of tandem mass spectra for peptide identification. In *Proc. of the* 7^{th} *Annual International Conference on Research in Computational Molecular Biology (RECOMB)*, pages 8–19, 2003.

[10] A. Bairoch and B. Boeckmann. The SWISS-PROT protein sequence data bank. *Nucleic Acids Res.*, 20:2019–2022, 1992.

[11] R. Bakhtiar and R. W. Nelson. Mass spectrometry of the proteome. *Mol. Pharmacol.*, 60:405–415, 2001.

[12] M. A. Baldwin. Protein identification by mass spectrometry: issues to be considered. *Mol. Cell. Proteomics*, 3(1):1–9, 2004.

[13] N. Bansal, M. Cieliebak, and Zs. Lipták. Efficient algorithms for finding submasses in weighted strings. In *Proc. of the Fifteenth Annual Combinatorial Pattern Matching Symposium (CPM 2004)*, volume 3109 of *Lect. Notes Comp. Sci.*, pages 194–204. Springer, 2004.

[14] M. Beck, R. Diaz, and S. Robins. The Frobenius problem, rational polytopes, and Fourier-Dedekind sums. *J. Number Theory*, 96:1–21, 2002.

[15] M. Beck and I. M. Gessel. The polynomial part of a restricted partition function related to the Frobenius problem. *Electron. J. Comb.*, 8:E1–E5, 2001.

[16] G. Bennett. Probability inequalities for the sum of independent random variables. *J. Amer. Statist. Assoc.*, 57:33–45, 1962.

[17] G. Bennett. Upper bounds on the moments and probability inequalities for the sum of independent, bounded random variables. *Biometrika*, 52:559–569, 1965.

[18] P. Berndt, U. Hobohm, and H. Langen. Reliable automatic protein identification from matrix-assisted laser desorption/ionization mass spectrometric peptide fingerprints. *Electrophoresis*, 20(18):3521–3526, Dec 1999.

[19] P. Billingsley. *Probability and Measure*. Wiley Interscience, New York, 3rd edition, 1995.

[20] S. Böcker. Sequencing from compomers: Using mass spectrometry for DNA de-novo sequencing of 200+ nt. In *Proc. of the* 3^{rd} *International Workshop on Algorithms in Bioinformatics (WABI)*, pages 476–497, 2003.

[21] S. Böcker. SNP and mutation discovery using base-specific cleavage and MALDI-TOF mass spectrometry. *Bioinformatics, Supplement 1 (ISMB)*, pages i44–i53, 2003.

[22] S. Böcker. Sequencing from compomers: Using mass spectrometry for DNA de-novo sequencing of 200+ nt. *J. Comp. Biol.*, 11(6):1110–1134, 2004.

[23] S. Böcker and H.-M. Kaltenbach. Mass spectra alignments and their significance. In A. Apostolico, M. Crochemore, and K. Park, editors, *Combinatorial Pattern Matching*, volume 3537 of *Lect. Notes Comp. Sci.*, pages 429–441. Springer, 2005.

[24] S. Böcker and H.-M. Kaltenbach. Mass spectra alignments and their significance. *accepted for publication in J. Discr. Algorithms*, 2006.

[25] S. Böcker and Zs. Lipták. The money changing problem revisited: Computing the Frobenius number in time $O(ka_1)$. Technical Report 2004-02, Technische Fakultät der Universität Bielefeld, Abteilung Informationstechnik, 2004.

[26] S. Böcker and Zs. Lipták. Efficient mass decomposition. In *Proc. of ACM Symposium on Applied Computing (ACM SAC 2005)*, pages 151–157, Santa Fe, USA, 2005.

[27] S. Breen, M. S. Waterman, and N. Zhang. Renewal theory for several patterns. *J. Appl. Probab.*, 22:228–234, 1985.

[28] D. C. Chamrad, G. Körting, K. Stühler, H. E. Meyer, J. Klose, and M. Blüggel. Evaluation of algorithms for protein identification from sequence databases using mass spectrometry data. *Proteomics*, 4:619–628, 2004.

[29] T. Chen, M.-Y. Kao, M. Tepel, J. Rush, and G. M. Church. A dynamic programming approach to de novo peptide sequencing via tandem mass spectrometry. *J. Comp. Biol.*, 8(3):325–337, 2001.

[30] M. Cieliebak, T. Erlebach, Zs. Lipták, J. Stoye, and E. Welzl. Algorithmic complexity of protein identification: Combinatorics of weighted strings. *Discr. Appl. Mathem.*, 137(1):27–46, 2004.

[31] E. Cinlar. Markov additive processes I. *Z. Wahrscheinl. verw. Geb.*, 24:85–93, 1972.

[32] E. Cinlar. Markov additive processes II. *Z. Wahrscheinl. verw. Geb.*, 24:95–121, 1972.

[33] K. R. Clauser, P. Baker, and A. L. Burlingame. Role of accurate mass measurement (±10 ppm) in protein identification strategies employing MS or MS/MS and database searching. *Anal. Chem.*, 71:2871–2882, 1999.

[34] S. L. Cohen and B. T. Chait. Influence of matrix solution conditions on the maldi-ms analysis of peptides and proteins. *Anal. Chem.*, 68:31–37, 1996.

[35] J. Colinge, A. Masselot, M. Giron, T. Dessingy, and J. Magnin. OLAV: Towards high-throughput tandem mass spectrometry data identification. *Proteomics*, 3:1454–1463, 2003.

[36] J. Colinge, A. Masselot, and J. Magnin. A systematic statistical analysis of ion trap tandem mass spectra in view of peptide scoring. In *Proc. of the* 3^{rd} *International Workshop on Algorithms in Bioinformatics (WABI)*, volume 2812 of *Lect. Notes Comp. Sci.*, pages 25–38. Springer, 2003.

[37] K. R. Coombes, S. Tsavachidis, J. S. Morris, K. A. Baggerly, M.-C. Hung, and H. M. Kuerer. Improved peak detection and quantification of mass spectrometry data acquired from surface-enhanced laser desorption and ionization by denoising spectra with the undecimated discrete wavelet transform. *Proteomics*, 5(16):4107–4117, Nov 2005.

[38] V. Dančík, T. A. Addona, K. R. Clauser, J. E. Vath, and P. A. Pevzner. De-novo peptide sequencing via tandem mass spectrometry. *J. Comp. Biol.*, 6(3/4):327–342, 1999.

[39] Q. Ding, L. Xiao, S. Xiong, Y. Jia, H. Que, Y. Guo, and S. Liu. Unmatched masses in peptide mass fingerprints caused by cross-contamination: An updated statistical result. *Proteomics*, 3:1313–1317, 2003.

[40] B. Domon and R. Aebersold. Mass spectrometry and protein analysis. *Science*, 312:212–217, 2006.

[41] N. Edwards and R. Lippert. Generating peptide candidates from amino-acid sequence databases for protein identification via mass spectrometry. In *Proc. of the* 2^{nd} *International Workshop on Algorithms in Bioinformatics (WABI)*, pages 68–81, 2002.

[42] J. E. Elias, W. Haas, B. K. Fahery, and S. P. Gygi. Comparative evaluation of mass spectrometry platforms used in large-scale proteomics investigations. *Nat. Meth.*, 2(9):667–675, August 2005.

[43] J. K. Eng, A. L. McCormack, and J. R. Yates III. An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *J. Am. Soc. Mass Spectrom.*, 5:976–989, 1994.

[44] J. Eriksson and D. Fenyö. A model of random mass-matching and its use for automated significance testing in mass spectrometric proteome analysis. *Proteomics*, 2:262–270, 2002.

[45] W. Feller. *An Introduction to Probability Theory and its Applications*, volume I. John Wiley & sons, 3rd edition, 1968.

[46] J. B. Fenn, M. Mann, C. K. Meng, S. F. Wong, and C. M. Whitehouse. Electrospray ionization for mass spectrometry of large biomolecules. *Science*, 246:64–71, 1989.

[47] D. Fenyö and R. C. Beavis. A method for assessing the statistical significance of mass spectrometry-based protein identifications using general scoring schemes. *Anal. Chem.*, 75(4):768–774, 2003.

[48] D. Fenyö, J. Qin, and B. T. Chait. Protein identification using mass spectrometric information. *Electrophoresis*, 19:998–1005, 1998.

[49] A. Frank and P. Pevzner. PepNovo: de novo peptide sequencing via probabilistic network modeling. *Anal. Chem.*, 15:964–973, 2005.

[50] A. Ganapathy, X.-F. Wan, J. Wan, J. Thelen, D. W. Emerich, G. Stacey, and D. Xu. Statistical assessement for mass-spec protein identification using peptide fingerprinting approach. In *Proc. of the 26th Ann. Int. Conf. of the IEEE EMBS*, pages 3051–3054. IEEE, 2004.

[51] P. Gärdn, R. Alm, and J. Häkkinen. PROTEIOS: an open source proteomics initiativee. *Bioinf.*, 21:2085–2087, 2005.

[52] A. Gattiker, W. V. Bienvenut, A. Bairoch, and E. Gasteiger. FindPept, a tool to identify unmatched masses in peptide mass fingerprinting protein identification. *Proteomics*, 2:1435–1444, 2002.

[53] S. Gay, P.-A. Binz, D. F. Hochstrasser, and R. D. Appel. Modeling peptide mass fingerprinting data using the atomic composition of peptides. *Electrophoresis*, 20:3527–3534, 1999.

[54] S. Gay, P.-A. Binz, D. F. Hochstrasser, and R. D. Appel. Peptide mass fingerprinting peak intensity prediction: Extracting knowledge from spectra. *Proteomics*, 2:1374–1391, 2002.

[55] F. Gonnet, G. Lemaitre, G. Waksman, and J. Tortajada. MALDI/MS peptide mass fingerprinting for proteome analysis: identification of hydrophobic proteins attached to eucaryote keratinocyte cytoplasmic membrane using different matrices in concert. *Proteome Sci.*, 1:E1–E7, 2003.

[56] R. Gras, M. Müller, E. Gasteiger, S. Gay, P.-A. Binz, W. Bienvenut, C. Hoogland, J.-C. Sanchez, A. Bairoch, D. F. Hochstrasser, and R. D. Appel. Improving protein identification from peptide mass fingerprinting through a parametrized multi-level scoring algorithm and an optimized peak detection. *Electrophoresis*, 20:3535–3550, 1999.

[57] M. Gribskov and N. L. Robinson. Use of the receiver operating characteristic (ROC) analysis to evaluate sequence matching. *Computers Chem.*, 20(1):25–33, 1996.

[58] J. H. Gross. *Mass Spectrometry*. Springer-Verlag Berlin Heidelberg, 2004.

[59] S. Hahner, H.-C. Lüdemann, F. Kirpekar, E. Nordhoff, P. Roepstorff, H.-J. Galla, and F. Hillenkamp. Matrix-assisted laser desorption/ionization mass spectrometry (MALDI) of endonuclease digests of RNA. *Nucleic Acids Res.*, 25:1957–1964, 1997.

[60] W. A. Harris, D. J. Janecki, and J. P. Reilly. Use of matrix clusters and trypsin autolysis fragments as mass calibrants in matrix-assisted laser desorption/ionization time-of-flight mass spectrometry. *Rapid Commun. Mass Spectrom.*, 16:1714–1722, 2002.

[61] M. Havilio, Y. Haddad, and Z. Smilansky. Intensity-based statistical scorer for tandem mass spectrometry. *Anal. Chem.*, 75:435–444, 2003.

[62] W. J. Henzel, T. M. Billeci, J. T. Stults, S. C. Wong, C. Grimley, and C. Watanabe. Identifying proteins from two-dimensional gels by molecular mass searching of peptide fragments in protein sequence databases. *Proc. Natl. Acad. Sci. USA*, 90(11):5011–5015, 1993.

[63] W. J. Henzel, C. Watanabe, and J. T. Stults. Protein identification: The origins of peptide mass fingerprints. *J. Am. Soc. Mass Spectrom.*, 14:931–942, 2003.

[64] H. Hermjakob, R. Giegerich, and W. Arnold. RIFLE: Rapid identification of microorganisms by fragment length evaluation. In *Proceedings of the* 5^{th} *International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pages 131–139, Halkidiki, Greece, June 1997.

[65] R. Higdon, J. M. Hogan, G. V. Belle, and E. Kolker. Randomized sequence databases for tandem mass spectrometry peptide and protein identification. *OMICS*, 9:364–379, 2005.

[66] F. Hillenkamp, M. Karas, R. C. Beavis, and B. T. Chait. Matrix-assisted laser desorption/ionization mass spectrometry of biopolymers. *Anal. Chem.*, 63(24):1193A–1203A, 1991.

[67] X. Huang and M. S. Waterman. Dynamic programming algorithms for restriction map comparison. *Comput. Appl. Biosci.*, 8(5):511–520, 1992.

[68] D. F. Hunt, J. R. Y. III, J. Shabanowitz, S. Winston, and C. R. Hauer. Protein sequencing by tandem mass spectrometry. *Proc. Natl. Acad. Sci. USA*, 83:6233–6237, 1986.

[69] P. James, M. Quadroni, E. Carafoli, and G. Gonnet. Protein identification by mass profile fingerprinting. *Biochem. Biophys. Res. Commun.*, 195(1):58–64, Aug 1993.

[70] L. H. Jeffery. *The Local Scripts of Archaic Greece.* Oxford University Press, 1961.

[71] R. S. Johnson and K. Biemann. Computer program (SEQPEP) to aid in the interpretation of high-energy collision tandem mass spectra of peptides. *Biomedical & Environmental mass spectrometry*, 18:945–957, 1989.

[72] H.-M. Kaltenbach, S. Böcker, and S. Rahmann. Markov additive chains and applications to fragment statistics for peptide mass fingerprinting. *Accepted for publication at RECOMB Satellite Workshop Systems Biology and Proteomics*, 2006.

[73] H.-M. Kaltenbach, H. Sudek, S. Böcker, and S. Rahmann. Statistics of cleavage fragments in random weighted strings. Technical Report 2005-06, Technische Fakultät der Universität Bielefeld, Abteilung Informationstechnik, 2005.

[74] H.-M. Kaltenbach, A. Wilke, and S. Böcker. SAMPI: Protein identification with mass spectra alignments. *Accepted for publication in BMC Bioinformatics*, 2007.

[75] M. Karas, U. Bahr, I. Fournier, M. Glückmann, and A. Pfenninger. The initial-ion velocity as a marker for different desorption-ionization mechanisms in MALDI. *Int. J. Mass Spec.*, 226:239–248, 2003.

[76] M. Karas and F. Hillenkamp. Laser desorption ionization of proteins with molecular masses exceeding 10,000 Daltons. *Anal. Chem.*, 60:2299–2301, 1988.

[77] J. A. Karty, M. M. Ireland, Y. V. Brun, and J. P. Reilly. Artifacts and unassigned masses encountered in peptide mass mapping. *J. Chromatogr. B*, 782:363–383, 2002.

[78] J. A. Karty, M. M. E. Ireland, Y. V. Brun, and J. P. Reilly. Defining absolute confidence limits in the identification of caulobacter proteins by peptide mass mapping. *J. Proteome Res.*, 1:325–335, 2002.

[79] A. Keller, A. I. Nesvizhskii, E. Kolker, and R. Aebersold. Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search. *Anal. Chem.*, 74(20):5383–5392, 2002.

[80] D. E. Knuth. *The Art of Computer Programming: Seminumerical Algorithms*, volume 2. Addison-Wesley, Reading, Massachusetts, third edition, 1997.

[81] E. Lange, C. Gröpl, K. Reinert, O. Kohlbacher, and A. Hildebrandt. High-accuracy peak picking of proteomics data using wavelet techniques. *Pac Symp Biocomput*, 2006.

[82] F. Levander, T. Rögnvaldsson, J. Samuelsson, and P. James. Automated methods for improved protein identification by peptide mass fingerprinting. *Proteomics*, 4:2594–2601, 2004.

[83] H. Lim, J. K. Eng, J. R. Yates III, S. L. Tollaksen, C. S. Giometti, J. F. Holden, M. W. W. Adams, C. I. Reich, G. J. Olsen, and L. G. Hays. Identification of 2D-gel proteins: a comparison of MALDI/TOF peptide mass mapping to mu LC-ESI tandem mass spectrometry. *J. Am. Soc. Mass. Spectrom.*, 14(9):957–970, 2003.

[84] F. Lisacek, S. Cohen-Boulakia, and R. D. Appel. Proteome informatics II: Bioinformatics for comparative proteomics. *Proteomics*, 6(20):5445–5466, 2006.

[85] H. Lodish, A. Berk, S. L. Zipursky, P. Matsudaira, D. Baltimore, and J. E. Darnell. *Molecular Cell Biology.* WH Freeman and Company: New York, NY, 5th ed. edition, 2004.

[86] P. G. Lokhov, O. V. Tikhonova, S. A. Moshkovskii, E. I. Goufman, M. V. Serebriakova, B. I. Maksimov, I. Y. Toropyguine, V. G. Zgoda, V. M. Govorun, and A. I. Archakov. Database search post-processing by neural network: Advanced facilities for identification of components in protein mixtures using mass spectrometric peptide mapping. *Proteomics*, 4:633–642, 2004.

[87] B. Lu and T. Chen. A suffix tree approach to the interpretation of tandem mass spectra: Applications to peptides of non-specific digestion and post-translational modifications. *Bioinformatics, Supplement 2 (ECCB)*, 19:ii113–ii121, 2003.

[88] B. Lu and T. Chen. Algorithms for de novo peptide sequencing using tandem mass spectrometry. *Drug Discov. Today*, 2:85–90, 2004.

[89] B. Ma, K. Zhang, C. Hendrie, C. Liang, M. Li, A. Doherty-Kirby, and G. Lajoie. PEAKS: Powerful software for peptide de novo sequencing by MS/MS. *Rapid Commun. Mass Spectrom.*, 17(20):2337–2342, 2003.

[90] M. Mann, R. C. Hendrickson, and A. Pandey. Analysis of proteins and proteomes by mass spectrometry. *Annu. Rev. Biochem.*, 70:437–473, 2001.

[91] M. Mann, P. Højrup, and P. Roepstorff. Use of mass spectrometric molecular weight information to identify proteins in sequence databases. *Biol. Mass Spectrom.*, 22(6):338–345, 1993.

[92] M. Mann and M. Wilm. Error-tolerant identification of peptides in sequence databases by peptide sequence tags. *Anal. Chem.*, 66(24):4390–4399, 1994.

[93] R. Matthiesen, M. Lundsgaard, K. Welinder, and G. Bauw. Interpreting peptide mass spectra by VEMS. *Bioinf.*, 19(6):792–793, 2003.

[94] J. S. Morris, K. R. Coombes, J. Koomen, K. A. Baggerly, and R. Kobayashi. Feature extraction and quantification for mass spectrometry in biomedical applications using the mean spectrum. *Bioinf.*, 21(9):1764–1775, 2005.

[95] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, 48:443–453, 1970.

[96] P. Ney and E. Nummelin. Markov additive processes I. Eigenvalue properties and limit theorems. *Ann. Probab.*, 15(2):561–592, 1987.

[97] P. Ney and E. Nummelin. Markov additive processes II. Large deviations. *Ann. Probab.*, 15(2):593–609, 1987.

[98] E. Nordhoff, A. Ingendoh, R. Cramer, A. Overberg, B. Stahl, M. Karas, F. Hillenkamp, and P. Crain. Matrix-assisted laser desorption/ionization mass spectrometry of nucleic acids with wavelengths in the ultraviolet and infrared. *Rapid Commun. Mass Spectrom.*, 6(23):771–776, 1992.

[99] E. Nordhoff, C. Luebbert, G. Thiele, V. Heiser, and H. Lehrach. Rapid determination of short DNA sequences by the use of MALDI-TOF. *Nucleic Acids Res.*, 28(20):7039–7044, 2000.

[100] J. V. Olsen, S.-E. Ong, and M. Mann. Trypsin cleaves exclusively c-terminal to arginine and lysine residues. *Mol. Cell. Proteomics*, 3.6:608–614, 2004.

[101] S.-E. Ong and M. Mann. Mass-spectrometry based proteomics turns quantitive. *Nat. chem. biol.*, 1:252–262, 2005.

[102] P. M. Palagi, P. Hernandez, D. Walther, and R. D. Appel. Proteome informatics I: Bioinformatics tools for processing experimental data. *Proteomics*, 6(20):5435–5444, 2006.

[103] D. J. C. Pappin, P. Hojrup, and A. J. Bleasby. Rapid identification of proteins by peptide-mass fingerprints. *Curr. Biol.*, 3(6):327–332, 1993.

[104] K. A. Parker. Scoring methods in MALDI peptide mass fingerprinting: ChemScore, and the ChemApplex program. *J. Am. Soc. Mass Spectrom.*, 13:22–39, 2002.

[105] T. Patzkill. *Proteomics*. Kluwer Academic Publishers, 2002.

[106] D. Perkins, D. J. C. Pappin, D. Creasy, and J. Cottrell. Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis*, 20:3551–3567, 1999.

[107] P. A. Pevzner, V. Dančík, and C. L. Tang. Mutation-tolerant protein identification by mass spectrometry. *J. Comp. Biol.*, 7(6):777–787, 2000.

[108] S. C. Pomerantz, J. A. Kowalak, and J. A. McCloskey. Determination of oligonucleotide composition from mass spectromectrically measured molecular weight. *J. Am. Soc. Mass Spectrom.*, 4:204–209, 1993.

[109] J. Quackenbush. Standardizing the standards. *Mol. Syst. Biol.*, 2:E1–E3, 2006.

[110] M. Régnier. A unified approach to word occurrence probabilities. *Discr. Appl. Mathem.*, 104:259–280, 2000.

[111] G. Reinert, S. Schbath, and M. S. Waterman. Probabilistic and statistical properties of words: An overview. *J. Comp. Biol.*, 7:1–46, 2000.

[112] S. Robin and J.-J. Daudin. Exact distribution of word occurrences in a random sequence of letters. *J. Appl. Probab.*, 36:179–193, 1999.

[113] S. Robin and J.-J. Daudin. Exact distribution of the distances between any occurrences of a set of words. *Ann. Inst. Statist. Math.*, 4:895–905, 2001.

[114] C. P. Rodi, B. Darnhofer-Patel, P. Stanssens, M. Zabeau, and D. van den Boom. A strategy for the rapid discovery of disease markers using the MassARRAY system. *BioTech.*, Suppl:62–69, Jun 2002.

[115] R. G. Sadygov and I. John R. Yates. A hypergeometric probability model for protein identification and validation using tandem mass spectral data and protein sequence databases. *Anal. Chem.*, 73:3792–3798, 2003.

[116] T. Sakurai, T. Matsuo, H. Matsuda, and I. Katakuse. PAAS 3: A computer program to determine probable sequence of peptides from mass spectrometric data. *Biomed. Mass Spec.*, 11:396–399, 1984.

[117] D. Sankoff and J. B. Kruskal. *Time Wraps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. Addison-Wesley, Reading, Mass., 1983.

[118] F. Schütz, E. A. Kapp, R. J. Simpson, and T. P. Speed. Deriving statistical models for predicting peptide tandem MS product ion intensities. *Biochem. Soc. Trans.*, 31(Pt 6):1479–1483, Dec 2003.

[119] R. Sedgewick and P. Flajolet. *An Introduction to the Analysis of Algorithms.* Addison-Wesley, 2nd edition, 2001.

[120] I. Shadforth, D. Crowther, and C. Bessant. Protein and peptide identification algorithms using MS for use in high-throughput, automated pipelines. *Proteomics*, 5:4082–4095, 2005.

[121] G. Siuzdak. *Mass spectrometry for biotechnology.* Academic Press, San Diego, 1996.

[122] A. P. Snyder. *Interpreting Protein Mass Spectra.* Oxford University Press, 2000.

[123] C. Tang, W. Zhang, D. Fenyö, and B. T. Chait. Assessing the performance of different protein identification algorithms. In *Proc. of the Am. Soc. Mass Spetrom. Conference*, 2000.

[124] J. A. Taylor and R. S. Johnson. Implementation and uses of automated de novo peptide sequencing by tandem mass spectrometry. *Anal. Chem.*, 73:2594–2604, 2001.

[125] Y. Wan, A. Yang, and T. Chen. PepHMM: A Hidden Markov Model based scoring function for mass spectrometry database search. *Anal. Chem.*, 78:432–437, 2006.

[126] M. S. Waterman. Estimating statistical significance of sequence alignments. *Phil. Trans. R. Soc. Lond. B*, 344:383–390, 1994.

[127] M. S. Waterman. *Introduction to Computational Biology.* CRC Press, Boca Raton, first edition, 1996.

[128] C. Wenk. Applying an edit distance to the matching of tree ring sequences in dendrochronology. In M. Crochemore and M. Paterson, editors, *Proceedings of Combinatorial Pattern Matching (CPM99)*, volume 1645 of *Lect. Notes Comp. Sci.*, pages 223–242, 1999.

[129] C. M. Whitehouse, R. N. Dreyer, M. Yamashita, and J. B. Fenn. Electrospray interface for liquid chromatographs and mass spectrometers. *Anal. Chem.*, 57(3):675–679, Mar 1985.

[130] H. Wilf. *generatingfunctionology.* Academic Press, 1990.

[131] A. Wilke, C. Rückert, D. Bartels, M. Dondrup, A. Goesmann, A. T. Hüser, S. Kespohl, B. Linke, M. Mahne, A. C. McHardy, A. Pühler, and F. Meyer. Bioinformatics support for high-throughput proteomics. *J. Biotechnol.*, 106(2–3):147–56, 2003.

[132] W. E. Wolski, M. Lalowski, P. Martus, R. Herwig, P. Giavalisco, J. Gobom, A. Sickmann, H. Lehrach, and K. Reinert. Transformation and other factors of the peptide mass spectrometry pairwise peak-list comparison process. *BMC Bioinformatics*, 6:E1–E21, 2005.

[133] A. J. Wyner. More on recurrence and waiting times. *Ann. Appl. Probab.*, 9:780–796, 1999.

[134] J. R. Yates III. Database searching using mass spectrometry data. *Electrophoresis*, 19(6):893–900, 1998.

[135] J. R. Yates III, J. K. Eng, and A. L. McCormack. Mining genomes: Correlating tandem mass-spectra of modified and unmodified peptides to sequences in nucleotide databases. *Anal. Chem.*, 67(18):3202–3210, 1995.

[136] J. R. Yates III, J. K. Eng, A. L. McCormack, and D. Schieltz. Method to correlate tandem mass spectra of modified peptides to amino acid sequences in the protein database. *Anal. Chem.*, 67:1426–1436, 1995.

[137] J. R. Yates III, S. Speicher, P. R. Griffin, and T. Hunkapillar. Peptide mass maps: A highly informative approach to protein identification. *Anal. Biochem.*, 214:397–408, 1993.

[138] N. Zhang, R. Aebersold, and B. Schwikowski. ProbID: A probabilistic algorithm to identify peptides through sequence database searching using tandem mass spectral data. *Proteomics*, 2:1406–1412, 2002.

[139] W. Zhang and B. T. Chait. ProFound: an expert system for protein identification using mass spectrometric peptide mapping information. *Anal. Chem.*, 72(11):2482–2489, 2000.