# Unipept
# Computational Exploration of Metaproteome Data

Bart Mesuere

# Unipept
# Computational Exploration of Metaproteome Data

**Bart Mesuere**

Promotor: Prof. Peter Dawyndt

Thesis submitted to obtain the degree of
Doctor of Science: Computer Science

2015-2016

UNIVERSITEIT
GENT

Department of Applied Mathematics,
Computer Science, and Statistics
Ghent University

# Table of Contents

# Preface

Het dankwoord, het begin van een boek, maar toch ook een beetje het einde. De meest gelezen pagina's van een doctoraat. Maar waarom eigenlijk? Een van de weinige stukjes tekst in dit boek waar een glimp kan worden opgevangen van de man of vrouw achter het doctoraat? De enige pagina's die begrijpbaar zijn voor iedereen? Omdat het boek er meestal mee begint en men daarna afhaakt? Of toch gewoon uit *zou ik vermeld worden*-nieuwsgierigheid?

Als doctorandus zijn dit misschien wel de moeilijkste pagina's om te schrijven. Je bent het beu en wil het boek naar de drukker sturen, maar tegelijk wil je niemand vergeten te bedanken, wat origineel uit de hoek komen en als het even kan ook wat grappig zijn. Ik doe zoals elke goede informaticus die vast zit met een probleem: Google openen. "*How to write an original preface*", "*Top 10 acknowledgements*", "*Auto generate acknowledgements*", "*Origin of Lorem Ipsum*", .... Weinig resultaat, maar zoals vaak wel twee uur tijd verspild op Wikipedia. Wist je trouwens dat de olifant, kangoeroe, en zeekoe de enige zoogdieren zijn die meerdere keren van gebit wisselen tijdens hun leven?

Leuke dankwoorden beginnen vaak met het bedanken van belangrijke personen die slechts indirect een grote invloed gehad hebben. Bij deze wens ik de Oromo te bedanken voor het ontdekken van de koffie (wist je trouwens dat het woord *coffee* van koffie komt, en niet andersom), Alan Turing voor zijn werk in de theoretische informatica (en misschien wel de allereerste bioinformaticus?), en professor Frank De Clerck die alles *triviaal* deed lijken. Ook schrijvende voorbeelden, merci Douglas Adams en Steven Moffat, en een sarcastische bedanking mogen uiteraard niet ontbreken. Bedankt aan mijn vroegere wiskundeleerkracht Dorine "informatica aan de unief dat zal niet lukken, en burgerlijk ingenieur zeker niet" Claeys.

Laat ik nu toch ook maar de mensen bedanken die er echt toe deden. In de eerste plaats denk ik dan aan mijn promotor Peter Dawyndt zonder wie dit doctoraat niet mogelijk was. Niettegenstaande proteomics voor ons beide onbekend terrein was aan het begin van dit project, zijn we er toch in geslaagd om op relatief korte termijn een mooie tool te ontwikkelen die ook effectief gebruikt wordt. Dit ook dankzij Peter Vandamme en Bart Devreese die ons, zeker in het begin, met veel geduld wegwijs gemaakt hebben in de uitdagingen en noden in dit toch wel complexe veld.

Eveneens een dikke merci aan de thesisstudenten die ik de afgelopen jaren begeleid heb: Toon, Tom, Felix, Kevin en Stijn. Jullie werk was een meerwaarde voor het project. Daarnaast hebben vele anderen, direct of indirect, bijgedragen aan het succes van Unipept. Bedankt Griet, Maarten, Lieven, Jens, Cizar, Henning en Lennart.

Leve ook het chatkanaal en de leden van de studentenvereniging Zeus WPI. Jullie expertise, tips, en feedback waren van onschatbare waarde en hebben me uren tijd bespaard. Jullie afleiding en interessante discussies hebben me dan weer uren tijd gekost.

Uiteraard wil ik ook mijn vakgroep TWIST bedanken die de werkdagen de moeite waard maakten dankzij de discussies in de koffiepauzes, de spelletjes- en K3-avonden, de filosofische gesprekken over mayonaise van de Aldi tijdens de restolunches en zo veel meer. Gaande van gedeelde interesses in Doctor Who, Zelda, Star Wars, Age of Empires, dino's en gekleurde bollen, over het leegspuiten van flessen cava en circulaire vergaderingen, tot het samen begeleiden van vakken, jullie verdienen het allemaal om hier vermeld te worden. Toch zou ik er graag enkele mensen uitpikken die er al van bij het begin bij waren. Een dikke merci aan mede-Tardisbouwer Davy, kousencontroleur Virginie, en alwetende Nico.

Jaren geleden belandde ik als 16-jarige op Home Boudewijn. Ik had toen nooit kunnen dromen in welke fantastische omgeving ik zou terecht komen. Tijdens de blok in de bar blijven slapen, illegaal netwerkkabels op het dak leggen, SABAM aan de deur zetten, frigo's uitkuisen met zoutzuur, het zijn maar enkele stoten die we uitgehaald hebben en de basis voor een vriendschap die nu, vele jaren later, nog verder blijft duren. Bedankt allemaal en dat er nog veel Ardennenreisjes mogen volgen. In het bijzonder nog een extra bedankje aan PhD-buddies Simon en Bert die altijd klaar stonden met advies en een luisterend oor waren als het even te veel werd.

Als laatste ook een grote dankjewel aan mijn grootste supporters, mijn familie. Jullie waren er altijd voor me, ook toen mijn eerste bachelor niet van een leien dakje liep, ik soms wat slechtgezind was tijdens de examens, of als ik weer eens in het weekend moest werken. Merci, mama, papa, Pieter, Koen, levende knuffelbeesten Twix, Boris, Jules, en uiteraard ook Cath. Bedankt voor alles.

Het dankwoord, het einde van een werk, maar toch ook een beetje een begin.

# Summary

Microorganisms drive most of the chemical transformations crucial to sustaining life on Earth. Their ability to inhabit almost every environmental niche proves they possess an incredible diversity of physiological capabilities. However, little is known about the majority of the millions of microbial species that are predicted to exists, given that we are able to grow only an estimated 1% of these organisms under lab conditions. The emerging disciplines of metagenomics and metaproteomics take advantage of the current generation of sequencing technologies to recover genetic material and active proteins directly from environmental samples. These new approaches provide us with a "new kind of microscope" that is revolutionizing our understanding of the diversity and ecology of environmental communities. However, the computational and statistical tools to analyze metagenomics and metaproteomics data are clearly lagging behind the developments in sequencing technology.

In this thesis, we present an online web portal called Unipept that combines advanced algorithms, novel statistical methods and interactive visualizations for the analysis of metagenomics and metaproteomics data sets. It will equip the new microscope with more powerful lenses, enabling researchers to better zoom in on who is living in complex environmental communities, what they are doing there and how they are doing it.

The first introductory chapter is aimed at bringing computer scientists up to speed with the necessary biochemical and (micro)biological background. Next, we present the

Unipept web application as a tool for the diversity analysis of complex metaproteome samples. Chapter 3 builds on that by introducing two new ways to access the analysis tools offered by Unipept: a web-based API and a set of command line tools.

Chapter 4 covers the peptidome analysis part of the Unipept ecosystem. The Unique Peptide Finder is introduced as a new way to discover unique peptides that can be used as biomarkers in targeted metaproteomics, while Peptidome Clustering can be used to investigate the relatedness of organisms. These two tools are built on the same foundations using advanced JavaScript features to offer interactive visualizations and high-performance client-side calculations.

Chapter 5 tells the story of Unipept from a computer scientist's point of view. The chapter reconstructs the development timeline of Unipept including the technical design choices and failed experiments.

The last chapter explores ongoing and potential future extensions to Unipept, including a new metagenomics pipeline, the addition of functional analysis features, and additions to the Unique Peptide Finder.

# Samenvatting

Micro-organismen zijn verantwoordelijk voor de meeste chemische transformaties die cruciaal zijn voor het behoud van het leven op Aarde. Hun vermogen om op bijna elke plaats op Aarde, hoe extreem ook, aanwezig te zijn, bewijst dat ze een ongelooflijke diversiteit aan fysiologische mogelijkheden bezitten. Er is echter weinig bekend over de meerderheid van de miljoenen soorten bacteriën die verondersteld worden te bestaan. We zijn namelijk in staat om naar schatting slechts 1% van alle bestaande bacteriën te kweken in laboratoriumomstandigheden. De opkomende metagenomics en metaproteomics disciplines kunnen gebruik maken van de huidige generatie sequeneringstechnologie om genetisch materiaal en actieve eiwitten rechtstreeks te bepalen vanuit omgevingsstalen. Deze nieuwe benaderingen bieden ons een “nieuw soort microscoop” die een revolutie teweeg brengt in ons inzicht in de diversiteit en ecologie van microbiële ecosystemen. De computationele en statistische tools om de grote hoeveelheden metagenomics en metaproteomics gegevens te analyseren blijven echter duidelijk achter op de snelle ontwikkelingen op vlak van sequeneringstechnologie.

In deze thesis stellen we de webapplicatie Unipept voor. Deze applicatie combineert geavanceerde algoritmen, nieuwe statistische methoden en interactieve visualisaties om metagenomics en metaproteomics data te analyseren. Unipept rust onze figuurlijke microscoop uit met nieuwe krachtige lenzen die onderzoekers toelaten om dieper in te zoomen op welke organismen aanwezig zijn in complexe ecosystemen, wat ze er doen en hoe ze dat doen.

Het eerste inleidende hoofdstuk probeert om informatici de nodige biochemische en (micro)biologische kennis bij te brengen om de rest van het onderzoek te begrijpen. Vervolgens introduceren we Unipept als een tool om de biodiversiteit van complexe omgevingsstalen te analyseren. Het derde hoofdstuk bouwt hierop verder door twee alternatieve manieren voor te stellen om toegang te krijgen tot de analyseresultaten van Unipept: een API en een commandolijn-toepassing.

Hoofdstuk 4 behandelt het peptidoomanalyseluik van Unipept. De *Unique Peptide Finder* is een nieuwe manier om unieke peptiden te ontdekken die kunnen gebruikt worden als biomerkers in *targeted* metaproteomics. De *Peptidome Clustering* toepassing kan gebruikt worden om de verwantschap tussen organismen te onderzoeken. Deze twee toepassingen gebruiken een gemeenschappelijke technische basis die steunt op geavanceerde JavaScript functies die een performante toepassing met interactieve visualisaties toelaten.

Hoofdstuk 5 vertelt het verhaal van Unipept vanuit het standpunt van een informaticus. Dit hoofdstuk reconstrueert de geschiedenis van de ontwikkeling van Unipept met aandacht voor de technische keuzes en mislukte experimenten.

Het laatste hoofdstuk beschrijft toekomstige uitbreidingen van Unipept. Voorbeelden hiervan zijn een nieuwe pipeline om metagenomics data te verwerken, de mogelijkheid om naast de biodiversiteit ook de functies van de eiwitten in kaart te brengen en enkele uitbreidingen op de *Unique Peptide Finder*.

# List of publications

Tsilia, Varvara, Bart Devreese, Ilse De Baenst, **Bart Mesuere**, Andreja Rajkovic, Mieke Uyttendaele, Tom Van De Wiele, and Marc Heyndrickx. 2012. "Application of MALDI-TOF mass spectrometry for the detection of enterotoxins produced by pathogenic strains of the Bacillus cereus group." *Analytical and Bioanalytical Chemistry* 404 (6-7): 1691–1702.

**Mesuere, Bart**, Bart Devreese, Griet Debyser, Maarten Aerts, Peter Vandamme, and Peter Dawyndt. 2012. "Unipept: Tryptic peptide-based biodiversity analysis of metaproteome samples." *Journal of Proteome Research* 11 (12): 5773–80.

**Mesuere, Bart**, Griet Debyser, Maarten Aerts, Bart Devreese, Peter Vandamme, and Peter Dawyndt. 2015. "The Unipept metaproteomics analysis pipeline." *Proteomics* 15 (8): 1437–42.

Debyser, Griet, **Bart Mesuere**, Lieven Clement, Jens Van de Weygaert, Pieter Van Hecke, Gwen Duytschaever, Maarten Aerts, Peter Dawyndt, Kris De Boeck, Peter Vandamme, and Bart Devreese. 2015. "Faecal proteomics: A tool to investigate dysbiosis and inflammation in patients with cystic fibrosis." *Journal of Cystic Fibrosis* 15 (2): 242–50.

**Mesuere, Bart**, Toon Willems, Felix Van der Jeugt, Bart Devreese, Peter Vandamme, and Peter Dawyndt. 2016. "Unipept web services for metaproteomics analysis." *Bioinformatics* (in press).

# Chapter 1

# Proteomics for computer scientists

Biology can be a daunting subject for computer scientists. As people who are used to everything being deterministic and logical, the real world can be disappointingly unpredictable. Biological lingo and jargon can also be a real barrier in the communication between computer scientists and biologists. This chapters aims to be a gentle introduction in the (micro)biology and biochemistry that is needed to understand this thesis.

## 1.1 Life on Earth

A good starting point to explain what life is, would be to start with its origin. Unfortunately, this origin of life is not something we know for sure. The origin of life is an active research area that tries to form a hypothesis that takes into account the biological, chemical as well as geophysical aspects.

### Spontaneous generation

Before the nineteenth century, it was generally believed that life could generate spontaneously out of non-living matter (Balme, 1962). Maggots, for example, could arise from dead flesh and crocodiles could form from logs rotting at the bottom of a pond according to the Greek philosopher Aristotle. During the seventeenth century, it gradually became clear that this belief was false and the theory that each living organism comes from a pre-existing living organism was adopted for visible organisms.

This theory is known as *spontaneous generation*.

At the time, there was a strong suspicion of the existence of organisms that aren't visible to the naked eye. In 1676, Antoni van Leeuwenhoek was the first to observe these microorganisms using a self-made microscope (Figure 1.1; Gest (2004)). The origin of these organisms was not clear and it wasn't until 1859, when Louis Pasteur did a series of famous experiments (Figure 1.2) that proved that microbial life couldn't spontaneously originate from a sterile nutrient broth, that the theory of spontaneous generation was refuted (Schwartz, 2001).

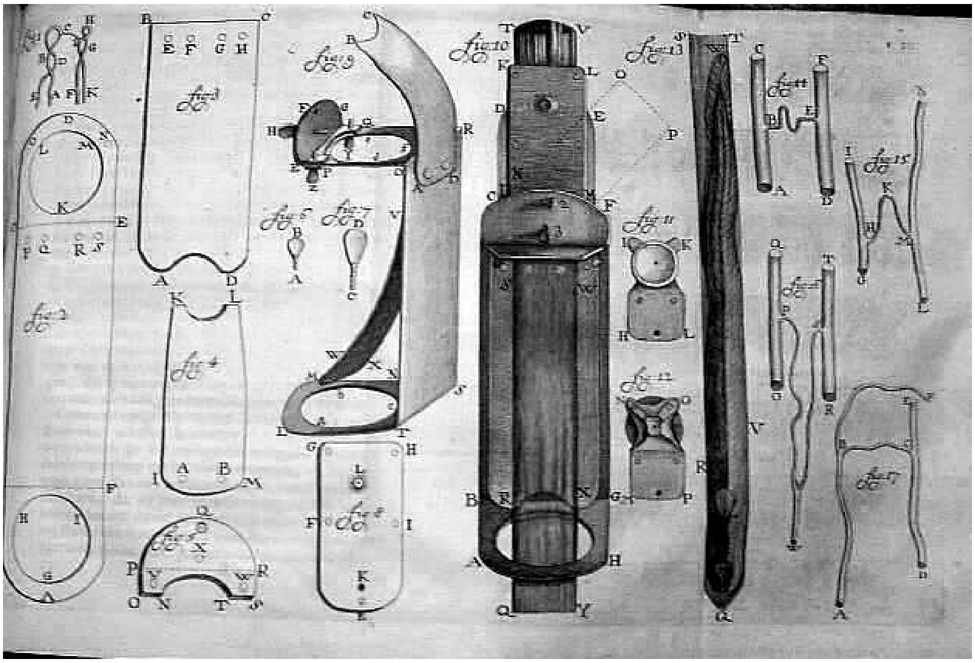


**Figure 1.1** Drawing from 1756 by English naturalist Henry Baker of microscopes owned by Antoni van Leeuwenhoek.

With spontaneous generation off the table, it took several years before a new theory was formulated. In 1924, Alexander Oparin speculated that the presence of oxygen in the atmosphere prevents the formation of the organic molecules that serve as building blocks for the evolution of life. This led Oparin to conclude that although spontaneous generation is not possible under the current circumstances, it did occur at least once a very long time ago when atmospheric oxygen was sparse. He argued that in an oxygen-less environment, a mix of organic compounds could indeed form by means of sunlight. These molecules could then cluster into more complex droplets by merging and splitting. This led to a sort of basic chemical evolutionary pressure that favors integrity (Oparin, 1953). Even today, Oparin's theory is still used as a starting point for many origin of life theories.

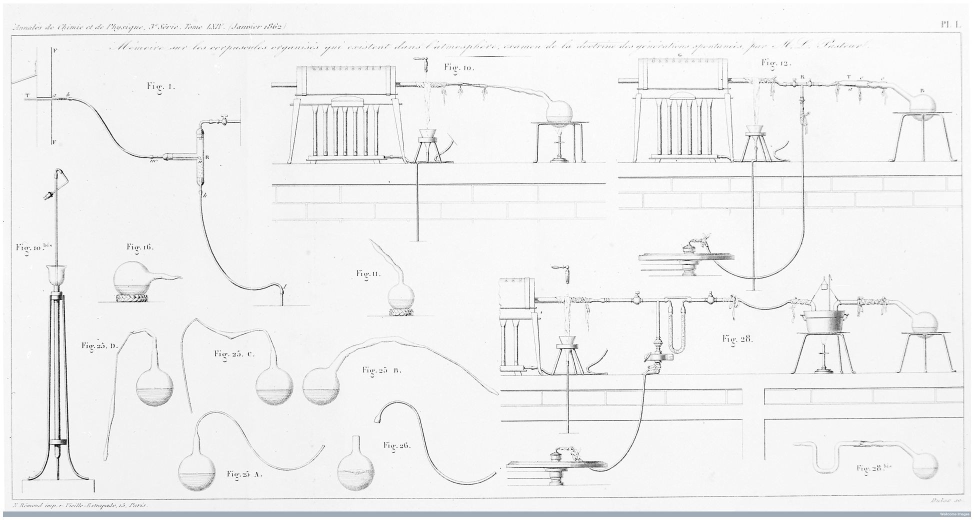


**Figure 1.2** Schematic of the experimental setup used by Louis Pasteur in his experiments on alleged spontaneous generation as published in *Mémoire sur les corpuscules organisés qui existent dans l'atmosphère: examen de la doctrine des générations spontanées* (Pasteur, 1862).

## Evolutionary timeline

To better understand its origin, it is worth taking a look at the oldest known life forms. One of the earliest evidences of life is found in the form of stromatolites (Garwood, 2012). Stromatolites are solid, rock-like structures that are formed by cyanobacteria, a type of aquatic bacteria that obtains energy through photosynthesis. Cyanobacteria can form colonies and capture sediments using a sticky, mucus-like surface layer. These sediments can react with water to form a thin layer of limestone. Over time, the limestone accumulates and forms a stromatolite (Riding, 1999). The earliest of these geological formations that contain presumed fossilized cyanobacteria date to over 3.5 billion years ago (Schopf et al., 2002). At that time, there was almost no free oxygen in Earth's atmosphere. By producing oxygen as a byproduct of converting carbon dioxide and water into sugar during photosynthesis, cyanobacteria contributed significantly to the oxygenation of the atmosphere. This led to

Stromatolite literally means *layered rock*.

the Great Oxygenation Event some 2.5 billion years ago (Flannery and Walter, 2012).

1.8 billion years ago, the first eukaryotic cells start to appear (Knoll et al., 2006). These are still unicellular organisms, but unlike prokaryotes, they contain cell organelles (such as a nucleus and mitochondria) enclosed by membranes. These organisms probably originated from several prokaryotic cells engulfing each other. Until then, the reproduction of eukaryotic organisms happened asexually through mitosis in which a cell divides into two genetically identical cells. Around 1.2 billion years ago, eukaryotic cells started to also reproduce sexually through meiosis (Bernstein and Bernstein, 2012). In meiosis, cell division produces daughter cells each containing half the genetic material of the parent cell. Two of these haploid cells can then fuse to create a new cell that contains the combined genetic material (Figure 1.3).

According to Butterfield (2000), sexual reproduction was critical for the success of the eukaryotes because it allowed for complex multicellularity. The first multicellular organism is believed to have originated more than 800 million years ago. 50 million years later, the first protozoa emerged, a group of unicellular eukaryotic organisms that exhibit advanced behavior such as motility and predation.

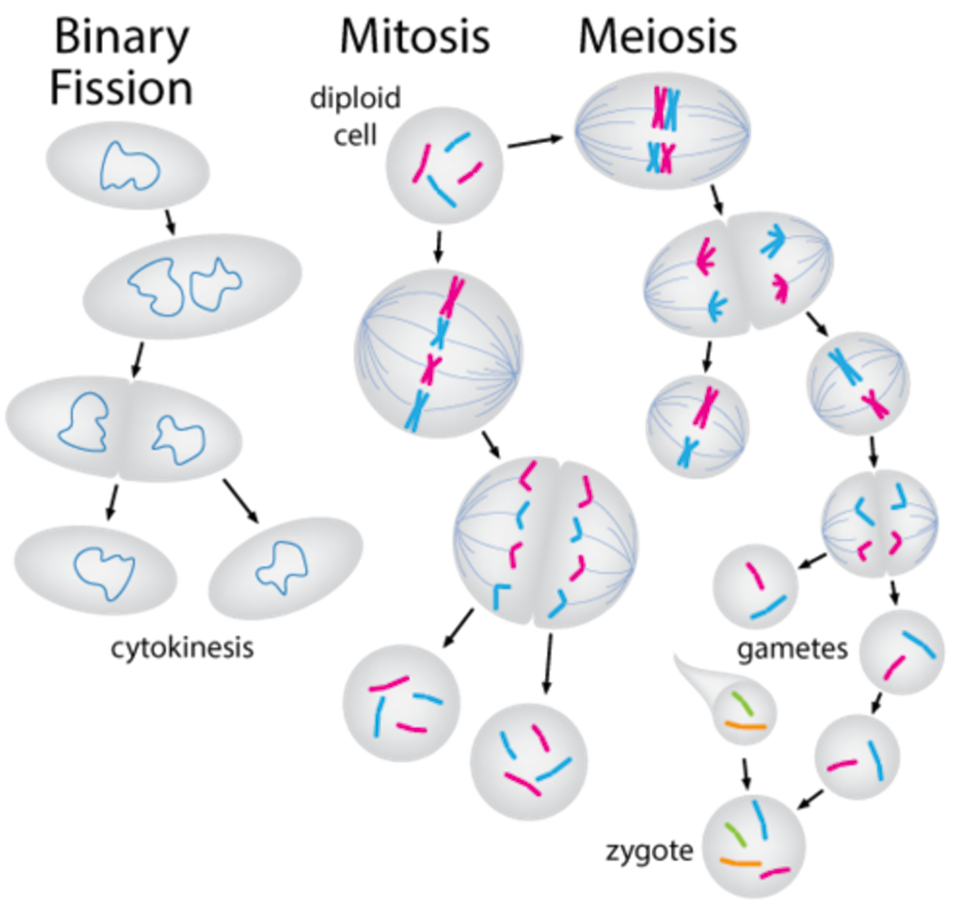


**Figure 1.3** Comparison between the different cell reproduction strategies. Binary fission is used by all prokaryotes and results into two identical parts. Mitosis and meiosis are cell division techniques used by eukaryotes with mitosis producing identical diploid cells and meiosis produces haploid cells. Image by the CK-12 Foundation under the CC BY-NC 3.0 license.

An interesting side effect of the rising atmospheric oxygen levels, is that high in the atmosphere, the oxygen molecules started interacting with each other under influence of the sun's ultraviolet (UV) radiation to form ozone molecules. Around 600 million years ago, a thin layer of ozone had built up around the Earth, protecting it from irradiation by the sun's UV light. Until then, life was limited to water, but the reduced radiation allowed for the colonization of the land (Battistuzzi, Feijao, and Hedges, 2004).

The combination of these events led to a period of increased evolutionary speed known as the Cambrian explosion. Starting around 580 million years ago for a period of 70 to 80 million years, the rate of diversification accelerated significantly (Marshall, 2006). Most of the major animal phyla appeared during that period such as jellyfish, crustaceans, arachnids, worms, etc. This period of rapid evolution was followed by the nascence of the first vertebrates (485 million years ago), the first primitive plants (434 million years ago), the first spiders and scorpions (420 million years ago), the first tetrapod on land (395 million years ago) and the first dinosaurs (225 million years ago; Lowe (2013)).

This event is known as the *Cretaceous-Paleogene extinction event*.

A massive comet impact some 66 million years ago had catastrophic effects on life on Earth: over 75 percent of all existing animal species was wiped out including all non-avian dinosaurs (Renne et al., 2013; Jablonski and Chaloner, 1994). The change in the environment and the eradication of many dominant groups let other organisms take their place. An example of this is the elimination of dinosaurs in favor of the mammals. One of these new types of mammals were the primates (60 million years ago) out of which the great apes (hominids; 18 million years ago) and eventually the humans (*Homo*; 2.5 million years ago) and modern humans (*Homo sapiens*; 250 000 years ago) formed (Goodman et al., 1998).

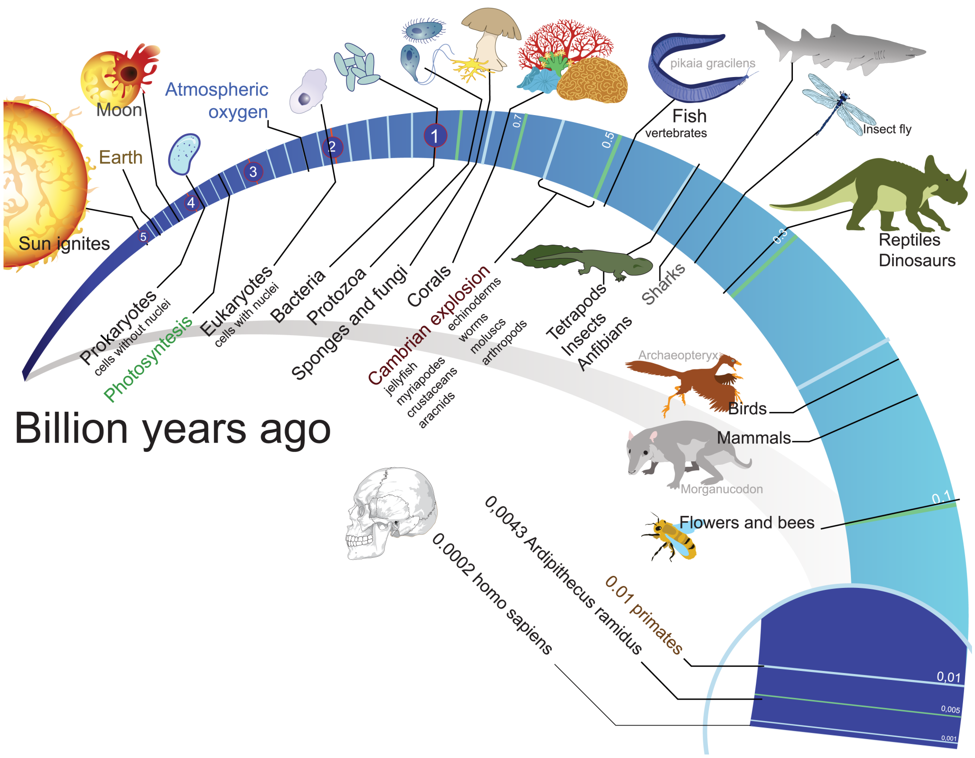


**Figure 1.4** Timeline showing the history of life on Earth.

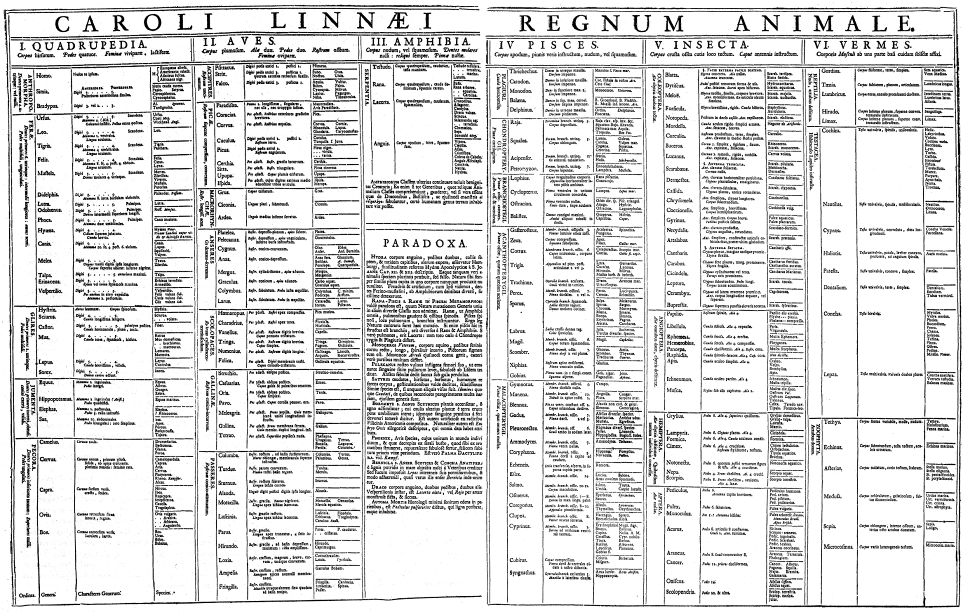
## 1.2 Taxonomy

Life on Earth is extremely varied, even more so than one would initially think. The group of beetles, for example, is incredibly diverse with over 400 000 species and gigantic compared to the number of mammals of which only 5 500 are known (Hammond, 1992). The number of described species easily exceeds 1 million and it is estimated that we have only managed to document a small fraction. Estimates for the total number of species range from 5 million to over 100 million. A recent statistical analysis estimates that the total number of non-bacterial species is 8.7 million (Mora et al., 2011).

Aristotle (384–322 BC) was one of the first to start naming and organizing living organisms (Mayr, 1982). He used a simple system with two groups: plants and animals. Organisms were put into classes based on their physical appearance and shape. This branch of science of naming and classifying organisms is called taxonomy and was relatively uneventful for the next 2000 years.

### Linnaean taxonomy

It wasn't until Carl Linnaeus (1707–1778), that taxonomy broke new ground. With the publication of the *Systema Naturæ* (Linnaeus, 1758), he introduced a standardized naming system for organisms. Next to a new naming system, he also introduced a new hierarchical classification system. In his taxonomy, there are three kingdoms (plants, animals, and minerals) that are each divided into several classes (Figure 1.5). These classes are then subdivided further in orders, families, genera, and species, each having their own name. The name of a species consists of two parts of which the first part refers to the parent genus. Although his system had numerous flaws, *minerals* can hardly be called living organisms and his class of *vermes* was a grab bag of organisms fitting nowhere else, the basic ideas of the Linnaean system are still used in today's systems.



CAROLI LINNÆI REGNUM ANIMALE.

I. QUADRUPEDIA. *Corpus hirsutum. Pedes quatuor. Feminæ viviparæ, lactiferæ.*

II. AVES. *Corpus plumosum. Alæ duæ. Pedes duo. Rostrum osseum. Feminæ oviparæ.*

III. AMPHIBIA. *Corpus nudum, vel squamosum. Dentes molares nulli: reliqui semper. Pinnæ nullæ.*

IV. PISCES. *Corpus apodum, pinnis veris instructum, nudum, vel squamosum.*

V. INSECTA. *Corpus crusta ossea cutis loco tectum. Caput antennis instructum.*

VI. VERMES. *Corporis Musculi ab una parte basi cuidam solidæ affixi.*

PARADOXA.

[illegible]

**Figure 1.5** The classification of animals in the classes of quadrupedia (mammals), aves (birds), amphibia (amphibians), pisces (fish), insecta (insects) and vermes ("animals of slow motion, soft substance, able to increase their bulk and restore parts which have been destroyed, extremely tenacious of life, and the inhabitants of moist places.") as described in the *Systema Naturæ* (Linnaeus, 1758).

## Evolutionary taxonomy

When we talk about ancestors in the rest of this thesis, such as in *Lowest Common Ancestor*, we mean a higher node in the taxonomy tree and not an evolutionary ancestor.

Towards the end of the eighteenth century, the idea formed to translate the Linnaean taxonomy, a system that produced systematic lists, into a tree-like organization of plants and animals. After the publication of Charles Darwin's theory of evolution in *On the Origin of Species* (Darwin, 1859), it gradually became accepted that classification should reflect Darwin's principle of common descent. The ensuing evolutionary taxonomy resulted in the generation of a tree of life that also included known fossil groups. The recent advent of DNA sequencing and analysis completed the transition from a taxonomy based entirely on morphology to one based on phylogeny, i.e., evolutionary history.

These recent advances don't mean that there is a single, official taxonomy containing all species. On the contrary, each domain has its own classification system. The Angiosperm Phylogeny Group III system (Bremer et al., 2009), for example, is used for flowering plants and the List of Prokaryotic names with Standing in Nomenclature (LPSN) by Euzéby (1997) is the authority for prokaryotes. In our application, we don't wish to limit ourselves to a single domain, which is why we use the NCBI Taxonomy database (Federhen, 2012). The NCBI Taxonomy is a nomenclature and classification repository that contains organism names and taxonomic lineages for all sequences in the databases of the International Nucleotide Sequence Database Collaboration (INSDC). Since the data used in our application also originates from INSDC databases, cross-references are ubiquitous.

## 1.3 Molecular building blocks

Although life on Earth is incredibly diverse, all organisms share fundamental molecular mechanisms. In almost all organisms, the basic unit of energy is adenosine triphosphate (ATP), structural and functional roles are fulfilled by proteins, and DNA (and RNA) carries the genetic information. In this section, we will take a closer look at the latter two.

### DNA

Deoxyribonucleic acid, or DNA, is a molecule that contains most of the genetic information that is needed for the development and functioning of all living organisms. The molecule consists of a long chain of many nucleotides. As can be seen in Figure 1.6, each nucleotide is composed of three main parts: a phosphate group, a 5-carbon sugar (de-

Each of our chromosomes is a single DNA molecule.

oxyribose) and one of four nitrogen-containing bases: adenine (A), thymine (T), cytosine (C) or guanine (G). Since the phosphate group and sugar are the same for every nucleotide, a DNA molecule can be described by the sequence of its bases.

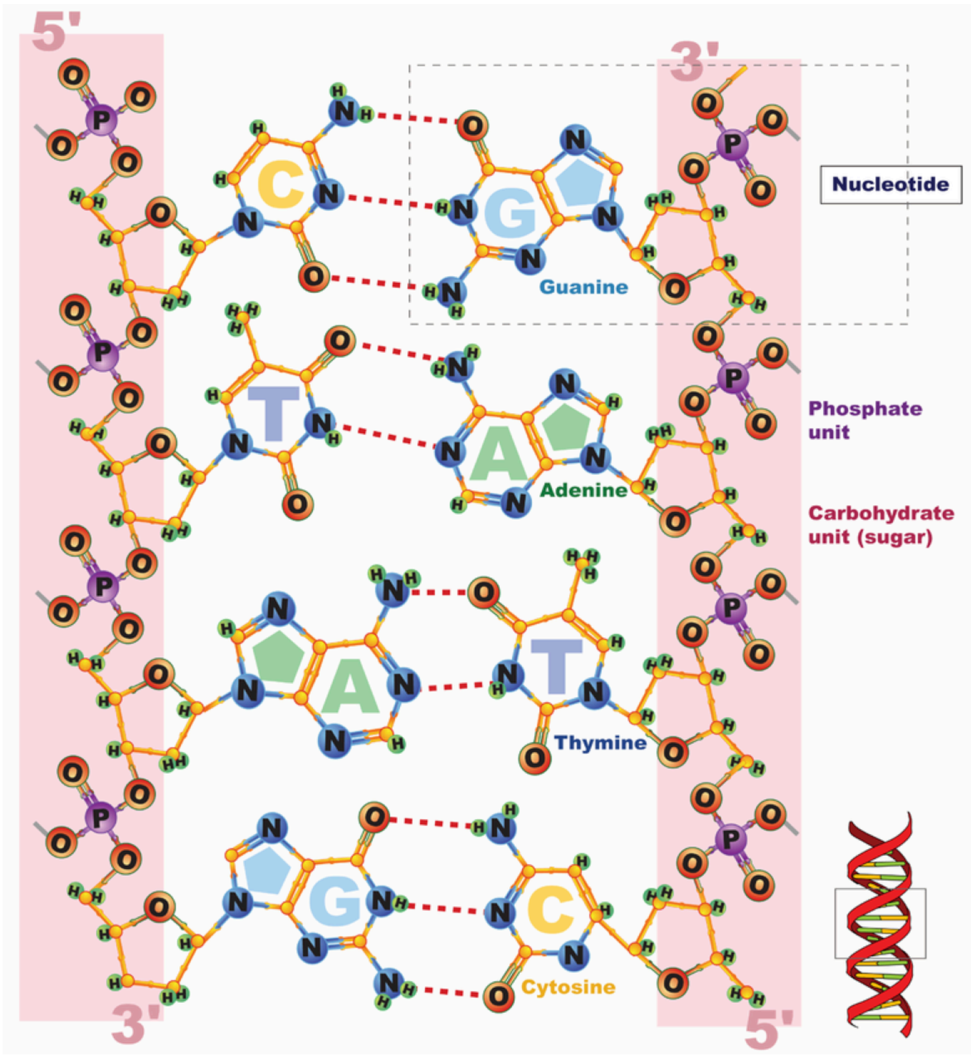


**Figure 1.6** The structure of a DNA molecule. Two complementing strands form a double helix structure. Each nucleotide consists of a deoxyribose sugar and a phosphate group at the outside of the helix and a nitrogen-containing base at the inside. The two complementing bases are joined with hydrogen bonds. Image by the CK-12 Foundation under the CC BY-NC 3.0 license.

In 1962, James Watson and Francis Crick (Figure 1.8) together with Maurice Wilkins won the Nobel Prize in Physiology or Medicine for their discovery of the molecular structure of DNA. Nine years before, they, together with Rosalind Franklin (Figure 1.7), determined that DNA is made of two strands of nucleotides that form a double helix (Watson and Crick, 1953). The nucleotides in the two-stranded spiral have their sugar and phosphate groups on the outside and their bases connecting on the inside. Not all bases can connect with each other: adenine always binds with thymine and cytosine always binds with guanine. This means that if one of the two strands is known, the complementary strand can be determined. For example, if the sequence of a strand contains ACCTGTC, the complementary section will be TGGACAG.

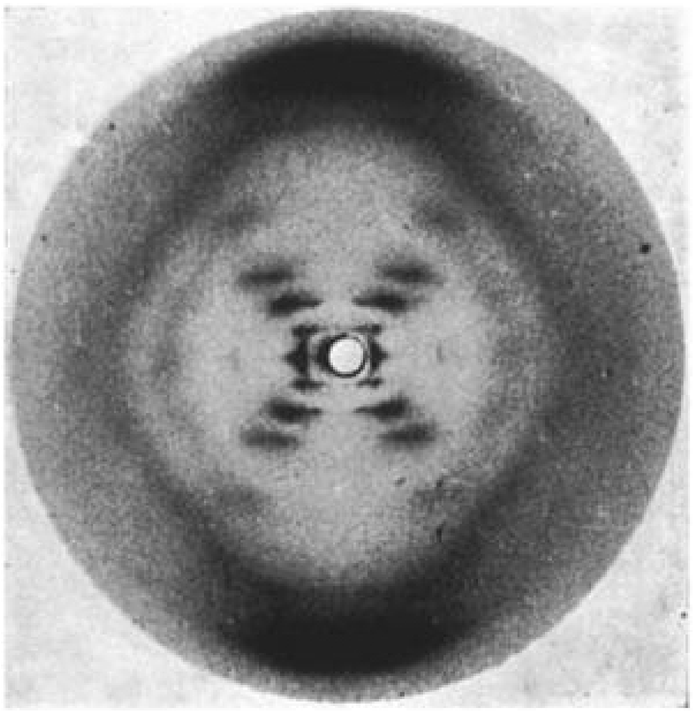


**Figure 1.7** "Photograph 51": X-ray diffraction image of crystallized DNA taken by Rosalind Franklin in 1953. The fuzzy X in the middle of the molecule indicates a helical structure.

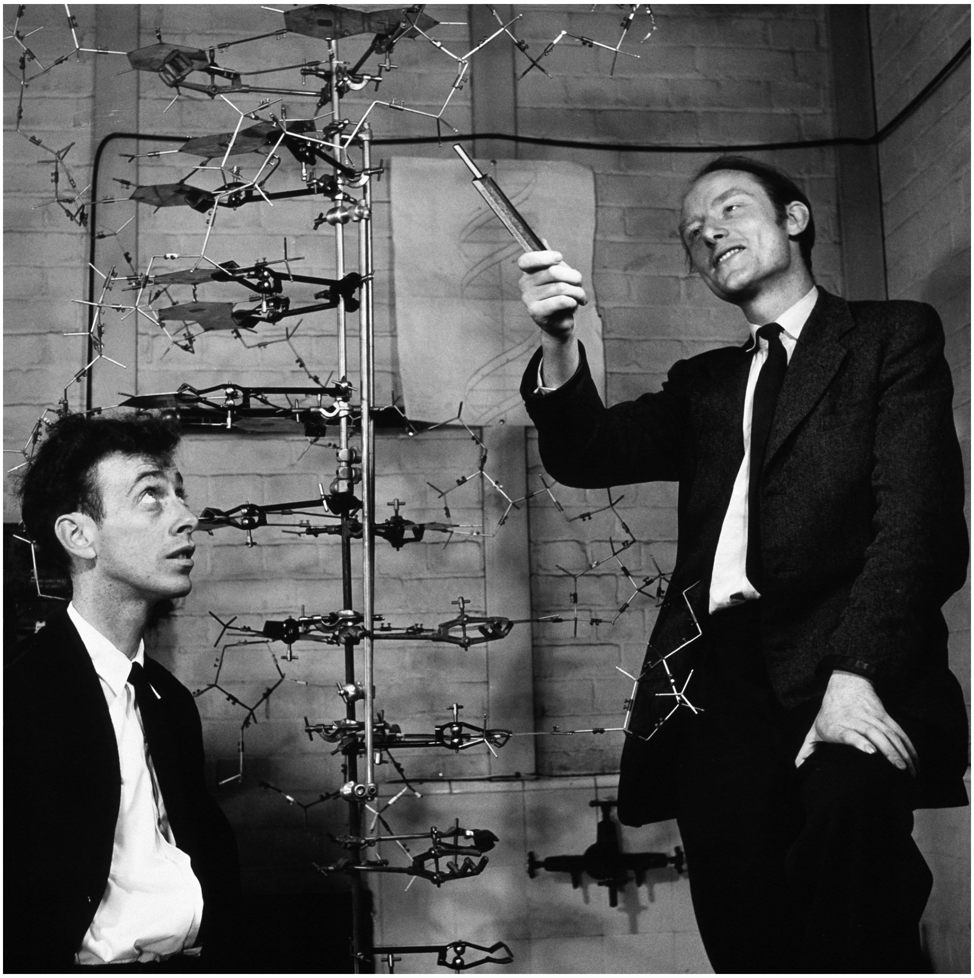


**Figure 1.8** James Watson and Francis Crick posing next to their double helix model in 1953. Photo taken by Antony Barrington Brown.

In eukaryotes, such as humans, DNA occurs in linear chromosomes while in most prokaryotes, such as bacteria, DNA occurs in circular chromosomes. All the chromosomes in the cell of an organism make up its genome. The size of the genome varies enormously across the tree of life. Viruses, for example, typically have a genome size of a few thousand base pairs (Fiers et al., 1976), the human genome has 3.2 billion base pairs spread across 46 chromosomes (Venter et al., 2001) and some plants have a genome of over 150 billion base pairs (Pellicer, Fay, and Leitch, 2010).

The genetic information itself is contained within genes. A gene is a part of the chromosome that encodes for a protein or a functional RNA. Not all of the DNA of an organism is part of a gene. In prokaryotes, 80-90% of the genome consists of coding DNA (Koonin and Wolf, 2010), but in eukaryotes this is many times lower. In humans, for example, over 98% of the genome is non-coding (Elgar and Vavouri, 2008). This non-coding DNA used to be called *junk DNA*, but recent research has shown that at least part of the non-coding DNA is biochemically active and performs regulatory functions (Pennisi, 2012).

## Gene expression

When a protein is made based on the information from a gene, we say that the gene is expressed. During the transcription phase of gene expression, a copy of the DNA sequence is made by RNA polymerase, creating messenger RNA. The protein-coding region of the messenger RNA is then translated into a protein (Figure 1.9).

As the name suggests, DNA uses deoxyribose, RNA uses ribose.

Ribonucleic acid (RNA) and DNA are both nucleic acids and share a lot of properties. Both are assembled as a chain of nucleotides, but unlike DNA, RNA mostly occurs as a single-strand. They both have the same phosphate group, but differ in sugar component and possible bases. In DNA, the complementing base of adenine is thymine, whereas in RNA, it is uracil (U).

Other types include small nuclear RNA and small interfering RNA.

Many types of RNA exist with the messenger RNA (mRNA), ribosomal RNA (rRNA) and transport RNA (tRNA) playing an important role in gene expression. The transcription phase starts by an RNA polymerase enzyme binding to the DNA molecule and opening up the double helix. The RNA polymerase then begins mRNA synthesis by matching bases that complement the DNA strand. If the DNA sequence is ATCCGA, for example, the resulting mRNA sequence will be UAGGCU. Once transcription is finished, the constructed mRNA is released in the cell.

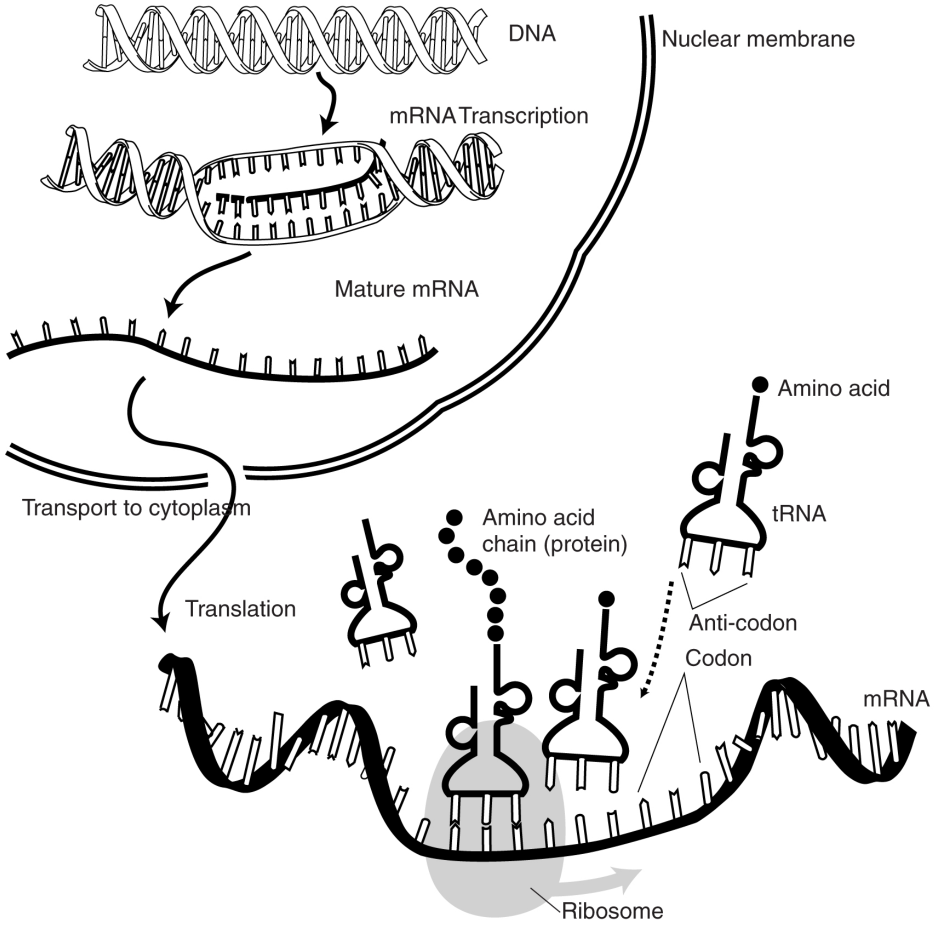


**Figure 1.9** Overview of gene expression in eukaryotes. DNA is first transcribed to messenger RNA in the nucleus. After transport out of the nucleus, protein synthesis begins with the help of a ribosome and transport RNA by translating nucleotide triplets to amino acids.

In the next phase, the mRNA binds to one of the ribosomes in the cell. A ribosome is a cell organelle that is responsible for translating the mRNA into a protein and consists of two subunits that are made from rRNA and proteins. The small subunit binds to the mRNA and reads the sequence. Each nucleotide triplet of the mRNA sequence is called a codon and can bind to a single type of transport RNA holding an amino acid. The large ribosomal subunit binds to the tRNA and connects the attached amino acid to the growing protein chain. The type of amino acid that is attached to the tRNA depends on the sequence of the codon binding site and is uniquely defined (Figure 1.10). This way, once the mRNA is bound to the ribosome, there is only a single protein translation possible.

### Proteins

Proteins are large molecules that perform many different functions in living organisms. They can serve as enzyme, have a structural function, or even transport other molecules. As mentioned in the previous section, the building blocks of proteins are amino acids. There are 20 naturally occurring amino acids, each having a three-letter and a one-letter abbreviation (Table 1.1). The one-letter abbreviations are also sometimes called the protein alphabet and are generally used to describe the sequence of the amino acids of a protein. Since proteins are formed by chains of amino acids, the order of them uniquely describes the protein. Shorter chains of amino acids are called peptides. The proteome is the set of all proteins that are expressed by an organism.

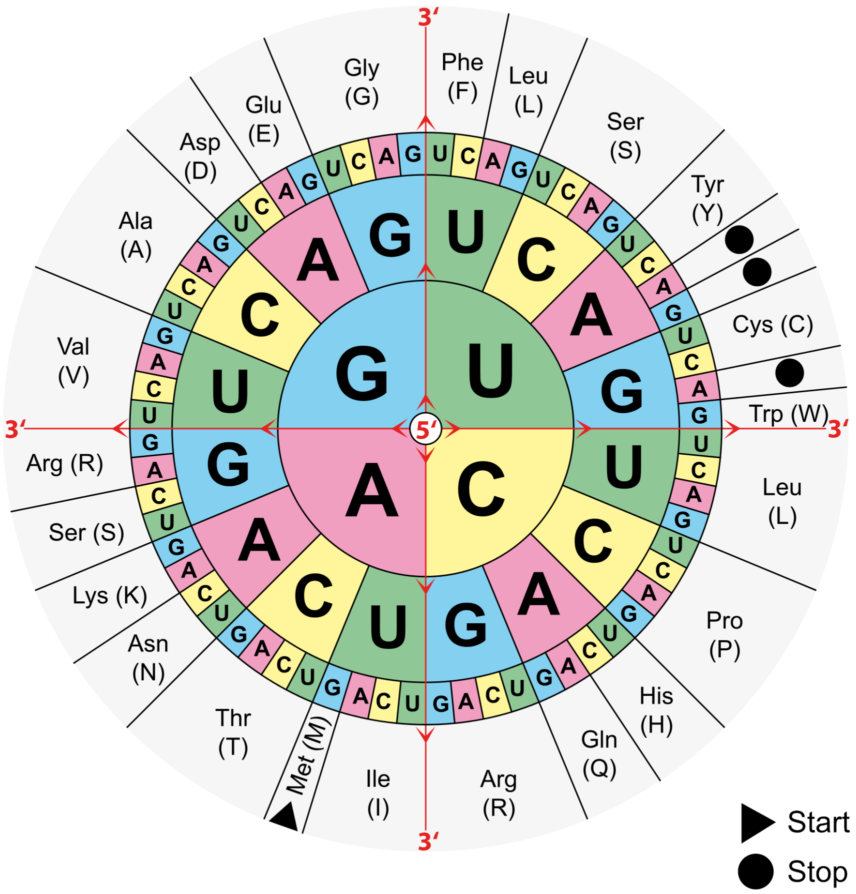


**Figure 1.10** An RNA codon table showing the mapping from three-letter RNA codons to amino acids. For example, the RNA codon AAC encodes the amino acid asparagine that is abbreviated as Asn or N. Different codons can result in the same amino acid, for example leucine has six possible codons, while tryptophan only has one.

**Table 1.1** Overview of the standard amino acids with long abbreviation, short abbreviation and molecular weight. Notice that leucine and isoleucine have the same weight.

| AMINO ACID | LONG ABBREVIATION | SHORT ABBREVIATION | AVERAGE MOLECULAR MASS |
|---|---|---|---:|
| Alanine | Ala | A | 89.09 Da |
| Arginine | Arg | R | 174.20 Da |
| Asparagine | Asn | N | 132.12 Da |
| Aspartic acid | Asp | D | 133.10 Da |
| Cysteine | Cys | C | 121.15 Da |
| Glutamic acid | Glu | E | 147.13 Da |
| Glutamine | Gln | Q | 146.15 Da |
| Glycine | Gly | G | 75.07 Da |
| Histidine | His | H | 155.16 Da |
| Isoleucine | Ile | I | 131.17 Da |
| Leucine | Leu | L | 131.17 Da |
| Lysine | Lys | K | 146.19 Da |
| Methionine | Met | M | 149.21 Da |
| Phenylalanine | Phe | F | 165.19 Da |
| Proline | Pro | P | 115.13 Da |
| Serine | Ser | S | 105.09 Da |
| Threonine | Thr | T | 119.12 Da |
| Tryptophan | Trp | W | 204.23 Da |
| Tyrosine | Tyr | Y | 181.19 Da |
| Valine | Val | V | 117.15 Da |

## 1.4 Metaproteomics

Examples of immunoassay techniques are ELISA and Western blot.

Proteomics is the large-scale study and analysis of proteins. The detection of specific proteins typically happens in two ways: using immunoassays and using mass spectrometry. The techniques using immunoassays use specific antibodies that bind to the target protein to detect and quantify that protein. Mass spectrometry on the other hand, is a more general technique that uses the mass of fragmentized ions to determine the chemical composition of a sample.

### Mass spectrometry

Mass spectrometry consists of three main parts: ionization, mass analysis and ion detection (Figure 1.11). During the ionization phase, a part of the sample is converted to ions. This means that the molecules are charged and fragmented by an ion source. Many different types of ion sources can be used depending on the type of sample that is subject to analysis. Two commonly used techniques for biological samples are electrospray ionization (ESI; Fenn et al. (1989)) and matrix-assisted laser desorption/ionization (MALDI; Tanaka et al. (1988)).

Next, the ions are selected from the sample and directed through the mass analyzer. This component uses electrical and/or magnetic fields to separate the ions based on their molecular mass and charge, or more specific, their mass-to-charge ratio. Again, many types of mass analyzers exist with time-of-flight (TOF) and Orbitrap (Hu et al., 2005) being common techniques. The combination of the used ion source and mass analyzer determines the configuration of a mass spectrometer. Common configurations often get their own name, for example, MALDI-TOF indicates that a MALDI ion source was used in combination with a TOF analyzer.

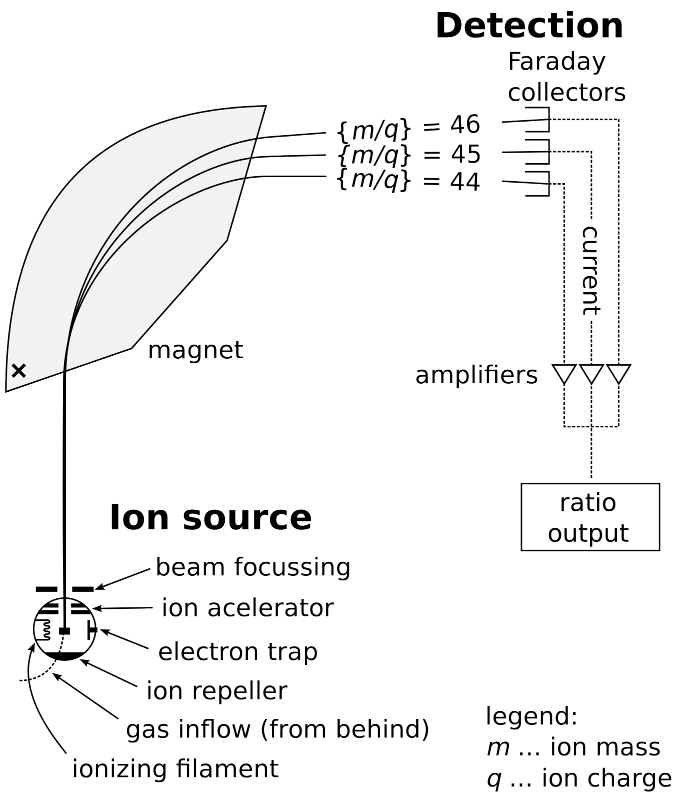


**Figure 1.11** Schematic overview of a mass spectrometer showing the ionization, mass analysis (magnet) and detector. Although this schematic gives a good idea about the internal process, the depicted (sector) instrument is not well suited for proteomics.

The separation of the ions makes it possible to detect how many ions of each mass-to-charge ratio (m/z) are present. This is what happens in the last phase by a detector producing a mass spectrum (Figure 1.12). This spectrum shows the number of detections and thus the relative abundance of ions for each mass-to-charge ratio.

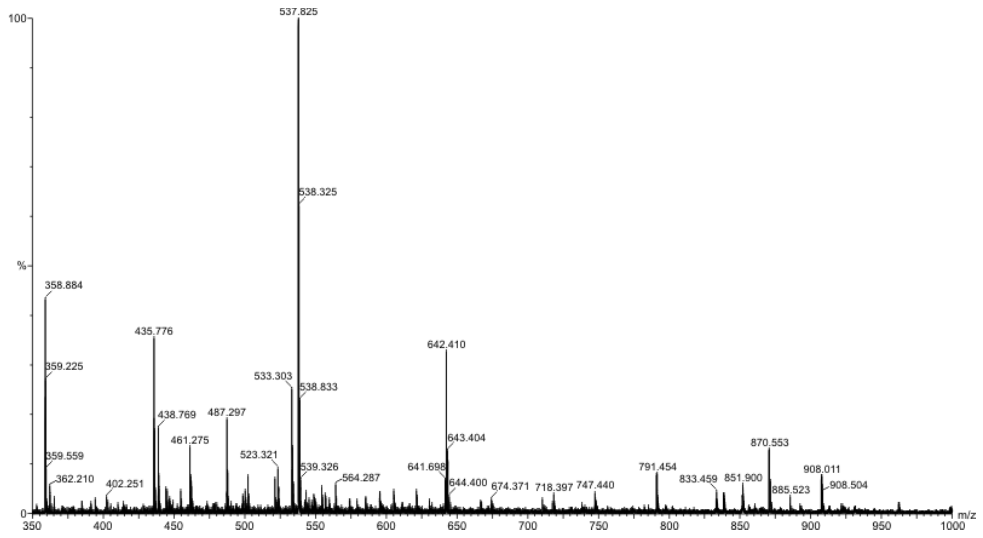


**Figure 1.12** Example of a recorded mass spectrum. The horizontal axis shows the recorded m/z range and the vertical axis shows the relative number of detections for each ratio.

## Proteomics

When using mass spectrometry for proteomics, several experimental techniques (e.g., top-down proteomics, bottom-up proteomics, targeted proteomics, etc.) can be used depending on the goal of the experiment. In this introduction we will focus on the bottom-up approach, where proteins are first split into smaller peptides before administering them to the mass spectrometer. Cleaving a protein in smaller peptides is done by adding a special, proteolytic, enzyme to the protein sample.

A proteolytic enzyme is also called a protease.

One of the most used proteases is trypsin (Vandermarliere, Mueller, and Martens, 2013), an enzyme that is found in the digestive system of humans and many other vertebrates where it helps to digest proteins in food. Trypsin cleaves peptide chains at very specific places namely when encountering the amino acids lysine (K) or arginine (R), except when they are bound to proline (P). The resulting peptides are called tryptic peptides. The effect of trypsin on a protein

Although generally accepted, the proline exception is increasingly being questioned.

can easily be simulated *in silico*. This comes down to splitting the protein sequence after every `K` or `R` except when followed by a `P`, for example using a regular expression like `s/([KR])([^P])/\1\n\2/g`. Unfortunately, trypsin sometimes fails to split the protein at a cleavage site. This is called a missed cleavage and most analysis software takes account this possibility.

Next, one or more mass spectrometry (MS) steps are used to analyze the peptide mixture. When two MS steps are used, this is called tandem mass spectrometry (MS/MS; Figure 1.13). Before the first MS phase, the peptides are ionized using an ionization technique that causes little fragmentation, for example ESI or MALDI. Next, the ions are separated and ions of a particular mass-to-charge ratio are selected. In the second phase, the selected ions are fragmented, for example using collision-induced dissociation (CID). The resulting fragments are then again separated by mass-to-charge ratio and detected resulting in a mass spectrum.

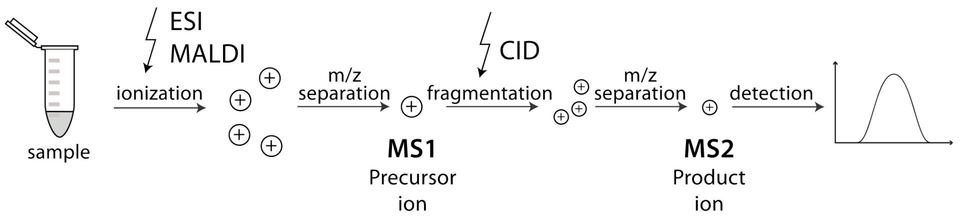


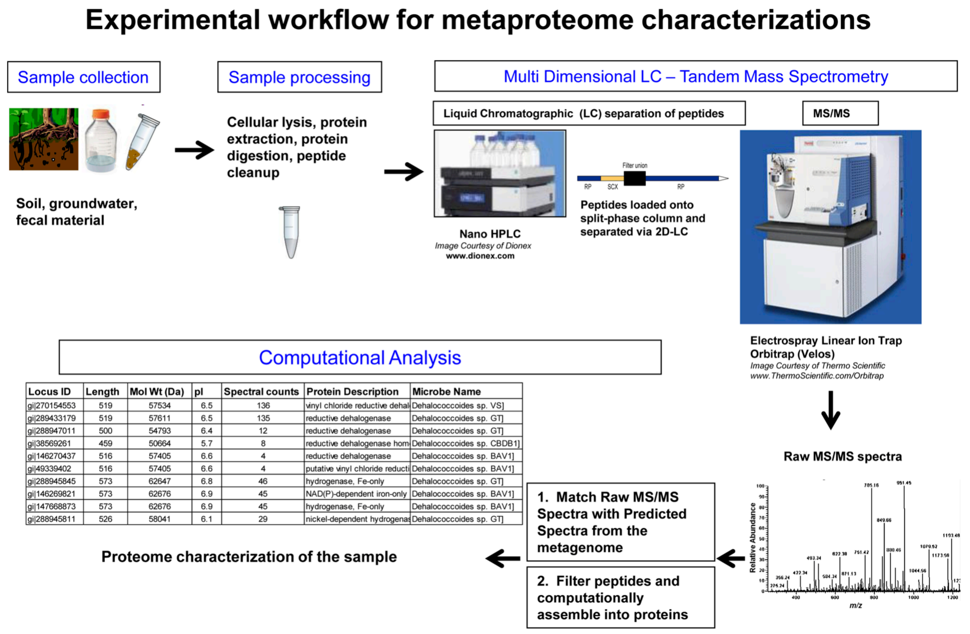
**Figure 1.13** Schematic overview of tandem mass spectrometry. In the first phase, the peptides are ionized (for example, using ESI or MALDI) and separated. In the second phase, ions of a particular mass-to-charge ratio are selected and fragmented using collision-induced dissociation (CID). Image by Hannes Röst under the CC BY-SA 3.0 license.

The next challenge is converting the many measured spectra to usable information such as peptide sequences. Two approaches can be used for this conversion: database searching and de novo sequencing. De novo sequencing uses advanced algorithms to determine the sequence directly from the spectrum. This allows the discovery of peptides that were never seen before, but the technique is time consuming and has a low accuracy (Pevtsov et al., 2006). Database searching is a simpler technique that is more commonly used. As the name implies, database searching starts from a protein database, for example UniProt, and performs an *in silico* trypsin digest on the proteins. For each of the resulting tryptic peptides, a theoretical mass spectrum is calculated. These predicted spectra are then compared to the recorded spectrum and the sequence of the best matching spectrum is returned as the result. Many tools implement database searching, for example Mascot (Hirosawa et al., 1993), Sequest (Eng, McCormack, and Yates, 1994), X!Tandem (Craig and Beavis, 2003) or OMSSA (Geer et al., 2004).

### Metaproteomics

Because an entire community is examined instead of an individual, metaproteomics is also sometimes called community proteomics.

When the protein contents of an environmental sample is analyzed, this is called metaproteomics. The origin of such samples ranges from waste water treatment plants to the human gut. While the procedure looks similar to the one described in the previous section, shotgun metaproteomics is a relatively new technique that developed in the last decade (Wilmes and Bond, 2004; Herbst et al., 2016). Because the origin of the proteins is not limited to a single species or well defined set of organisms, metaproteomics is a lot more challenging.


Experimental workflow for metaproteome characterizations
Sample collection
Soil, groundwater, fecal material
Sample processing
Cellular lysis, protein extraction, protein digestion, peptide cleanup
Multi Dimensional LC – Tandem Mass Spectrometry
Liquid Chromatographic (LC) separation of peptides
Nano HPLC
Image Courtesy of Dionex
www.dionex.com
Filter union
RP
SCX
RP
Peptides loaded onto split-phase column and separated via 2D-LC
MS/MS
Electrospray Linear Ion Trap Orbitrap (Velos)
Image Courtesy of Thermo Scientific
www.ThermoScientific.com/Orbitrap
Raw MS/MS spectra
Relative Abundance
m/z
Computational Analysis
Proteome characterization of the sample
1. Match Raw MS/MS Spectra with Predicted Spectra from the metagenome
2. Filter peptides and computationally assemble into proteins

| Locus ID | Length | Mol Wt (Da) | pI | Spectral counts | Protein Description | Microbe Name |
|---|---|---|---|---|---|---|
| gi\|270154553 | 519 | 57534 | 6.5 | 136 | vinyl chloride reductive dehal | Dehalococcoides sp. VS] |
| gi\|289433179 | 519 | 57611 | 6.5 | 135 | reductive dehalogenase | Dehalococcoides sp. GT] |
| gi\|288947011 | 500 | 54793 | 6.4 | 12 | reductive dehalogenase | Dehalococcoides sp. GT] |
| gi\|38569261 | 459 | 50664 | 5.7 | 8 | reductive dehalogenase hom | Dehalococcoides sp. CBDB1] |
| gi\|146270437 | 516 | 57405 | 6.6 | 4 | reductive dehalogenase | Dehalococcoides sp. BAV1] |
| gi\|49339402 | 516 | 57405 | 6.6 | 4 | putative vinyl chloride reducti | Dehalococcoides sp. BAV1] |
| gi\|288945845 | 573 | 62647 | 6.8 | 46 | hydrogenase, Fe-only | Dehalococcoides sp. GT] |
| gi\|146269821 | 573 | 62676 | 6.9 | 45 | NAD(P)-dependent iron-only | Dehalococcoides sp. BAV1] |
| gi\|147668873 | 573 | 62676 | 6.9 | 45 | hydrogenase, Fe-only | Dehalococcoides sp. BAV1] |
| gi\|288945811 | 526 | 58041 | 6.1 | 29 | nickel-dependent hydrogenas | Dehalococcoides sp. GT] |

**Figure 1.14** Example of a shotgun metaproteomics workflow used to identify proteins in environmental samples. Image taken from Hettich et al. (2013).

In classic proteomics, proteins are first separated and selected using two-dimensional gel electrophoresis (2DE) before trypsin is added to digest the proteins into tryptic peptides. In metaproteomics, we want to get a complete picture of the sample and analyze as many proteins as possible. Because of this, the 2DE step is usually skipped and trypsin is added in the sample preparation step (Figure 1.14). During sample preparation, the proteins are extracted from the cells in the sample. Protein extraction in complex samples is very challenging and the method used is dependent on the origin of the sample. Next, the tryptic peptides are separated using liquid chromatographic methods, for example high-performance liquid chromatography (HPLC), before starting the MS/MS phase.

Converting the obtained spectra to peptide sequences and ultimately proteins is also a lot more challenging in metaproteomics. Because the number of possible organisms and thus proteins is a lot bigger, the database on which a search is performed is very important and must be carefully selected (Tanca et al., 2013). Other issues that impede the correct identification of proteins is the fact that distinct peptides can generate the same spectrum (leucine and isoleucine have the same mass, for example) and that some peptides are shared by many proteins.

Nevertheless, the field of metaproteomics has seen enormous progress in the last ten years (Figure 1.15). Where the technique initially only worked for relatively simple acid mine drainage samples, it is now routinely applied to extremely complex environments such as the human gut.

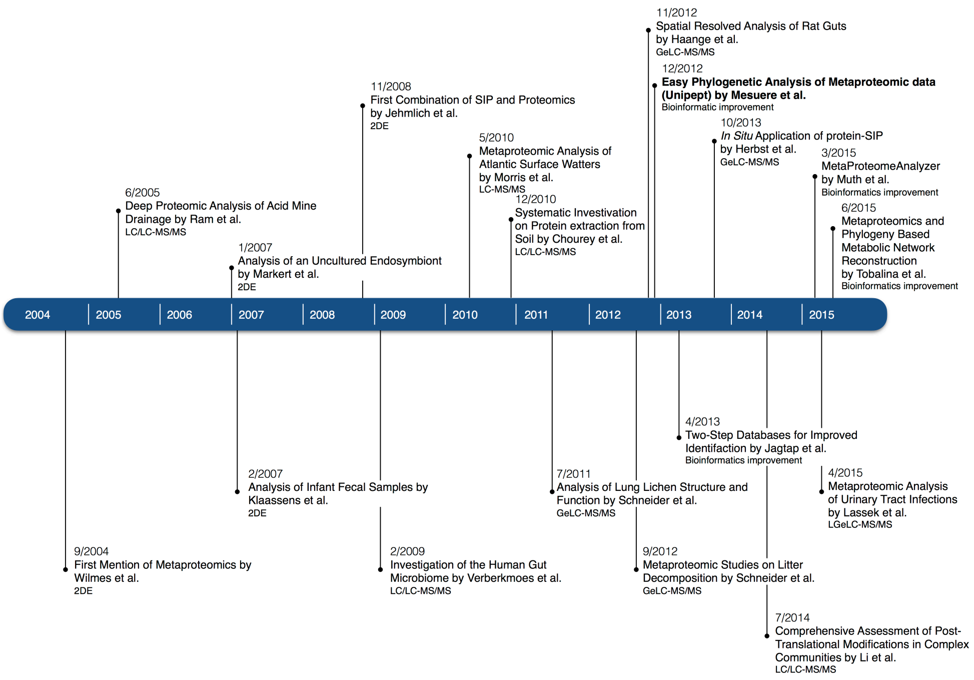

11/2012
Spatial Resolved Analysis of Rat Guts by Haange et al.
GeLC-MS/MS
12/2012
Easy Phylogenetic Analysis of Metaproteomic data (Unipept) by Mesuere et al.
Bioinformatic improvement
11/2008
First Combination of SIP and Proteomics by Jehmlich et al.
2DE
10/2013
In Situ Application of protein-SIP by Herbst et al.
GeLC-MS/MS
5/2010
Metaproteomic Analysis of Atlantic Surface Watters by Morris et al.
LC-MS/MS
3/2015
MetaProteomeAnalyzer by Muth et al.
Bioinformatics improvement
12/2010
Systematic Investivation on Protein extraction from Soil by Chourey et al.
LC/LC-MS/MS
6/2005
Deep Proteomic Analysis of Acid Mine Drainage by Ram et al.
LC/LC-MS/MS
6/2015
Metaproteomics and Phylogeny Based Metabolic Network Reconstruction by Tobalina et al.
Bioinformatics improvement
1/2007
Analysis of an Uncultured Endosymbiont by Markert et al.
2DE
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
4/2013
Two-Step Databases for Improved Identifaction by Jagtap et al.
Bioinformatics improvement
2/2007
Analysis of Infant Fecal Samples by Klaassens et al.
2DE
7/2011
Analysis of Lung Lichen Structure and Function by Schneider et al.
GeLC-MS/MS
4/2015
Metaproteomic Analysis of Urinary Tract Infections by Lassek et al.
LGeLC-MS/MS
9/2004
First Mention of Metaproteomics by Wilmes et al.
2DE
2/2009
Investigation of the Human Gut Microbiome by Verberkmoes et al.
LC/LC-MS/MS
9/2012
Metaproteomic Studies on Litter Decomposition by Schneider et al.
GeLC-MS/MS
7/2014
Comprehensive Assessment of Post-Translational Modifications in Complex Communities by Li et al.
LC/LC-MS/MS

**Figure 1.15** Overview of the milestones in metaproteomics research over the last ten years. Unipept is mentioned as bioinformatics improvement for the easy phylogenetic analysis of metaproteomic data. Image adapted from Herbst et al. (2016)

## 1.5 Unipept

Although identified proteins are usually the desired result of an MS experiment, we take a step back and start from the peptides.

Unipept, the subject of this thesis, is a tool to help analyze the outcome of metaproteomics experiments. More specifically, it takes the identified peptides as input and gives an overview of the biodiversity as output. This happens by exactly matching the peptides to the protein sequences in the UniProt database and aggregating the resulting associated organisms to a consensus taxon. Since UniProt contains over 60 million protein entries and a typical metaproteomics experiment yields thousands of identified peptides, significant time was spent to optimize the database and algorithms. This resulted in a web application that can analyze a sample containing thousands of identified peptides in mere seconds while presenting the results using rich and interactive visualizations.

In the next chapter, we present the Unipept web application as a tool for the diversity analysis of complex metaproteome samples. Chapter 3 builds on that by introducing two new ways to access the analysis tools offered by Unipept: a web-based API and a set of command line tools.

Chapter 4 covers the peptidome analysis part of the Unipept ecosystem. The Unique Peptide Finder is introduced as a new way to discover unique peptides that can be used as biomarkers in targeted metaproteomics, while Peptidome Clustering can be used to investigate the relatedness of organisms. These two tools are built on the same foundations using advanced JavaScript features to offer interactive visualizations and high-performance client-side calculations.

Chapter 5 tells the story of Unipept from a computer scientist's point of view. The chapter reconstructs the development timeline of Unipept including the technical design choices and failed experiments.

The last chapter explores ongoing and potential future extensions to Unipept, including a new metagenomics pipeline, the addition of functional analysis features, and additions to the Unique Peptide Finder.

# Chapter 2

# Metaproteomics biodiversity analysis

The initial objective of Unipept was to improve the biodiversity analysis of metaproteomics experiments. This was achieved not only by providing more accurate and faster results, but also by presenting the results with interactive visualizations in a user-friendly approach focused on non-technical users.

This chapter contains the initial Unipept article published in the Journal of Proteome Research in 2012, an update article published in Proteomics in 2015 and a short overview of the new features that were added since the last publication.

## 2.1 Unipept: Tryptic peptide-based biodiversity analysis of metaproteome samples

This section contains a verbatim copy of the research article by Mesuere et al. (2012) as published in the Journal of Proteome Research.

**Abstract** — The Unipept web application (http://unipept.ugent.be) supports biodiversity analysis of large and complex metaproteome samples using tryptic peptide information obtained from shotgun MS/MS experiments. Its underlying index structure is designed to quickly retrieve all occurrences of a tryptic peptide in UniProt entries. Taxon-specificity of the tryptic peptide is successively derived from these occurrences using a novel lowest common ancestor approach that is robust against taxonomic misarrangements, misidentifications, and inaccuracies. Not taking into account this identification noise would otherwise result in drastic loss of information. Dynamic treemaps visualize the biodiversity of metaproteome samples, which eases the exploration of samples with highly complex compositions. The potential of Unipept to gain novel insights into the biodiversity of a sample is evaluated by reanalyzing publicly available metaproteome data sets taken from the bacterial phyllosphere and the human gut.

### 2.1.1 Introduction

The introduction of high throughput sequencing methods allowed to determine the diversity, phylogeny, and genomic repertoire of complex microbial communities such as the human gut microbiome. Recently, the Metahit consortium released metagenomic sequence information showing approximately 1 000 different species commonly found in fecal samples, on average accounting for half a million genes in addition to the human genome (Qin et al., 2010). While metagenomics provides a wealth of information on the global gene content, understanding the actual functional contribution to nutrient conversion or immune system development of individual genes or organisms requires functional genomics tools.

High quality multidimensional liquid chromatography in combination with shotgun tandem mass spectrometric methods are currently implemented to reveal the protein complement of the metagenome, providing information of the core functional components (Verberkmoes et al., 2009; Kolmeder et al., 2012). In a typical single species proteomic experiment, protein identification from shotgun MS/MS data of tryptic peptides relies on matching the spectra to *in silico* calculated spectral information from proteins predicted from isolate or metagenomic databases. Therefore, tryptic digests and fragmentation ions on all protein sequences available for this organism are simulated. In the worst case, protein identification can be based on cross-species identification, typically using a close homologue.

In metaproteomics approaches, MS/MS based identification is hampered by several aspects. A first problem is the limited coverage of the curated protein databases, e.g., UniProtKB/Swiss-Prot (Boeckmann et al., 2003). Ideally, a protein complement of a synthetic metagenomic database containing sequences from different metagenomics experiments covering a wide range of organisms expected in the environment of interest could be created. Metagenomic databases however are exponentially increasing, and naive six-frame translation and protein prediction would lead to a high false discovery rate or low protein identification efficiency. Rooijers et al. (2011) countered this problem by implementing an iterative workflow combining the use of a defined synthetic metagenome and a non-annotated metagenome repository.

The taxon-specificity of peptides is discussed in more detail in Chapter 4.

A more specific problem towards functional analysis of the metaproteome is the lack of connectivity of the tryptic peptides and the organism of origin. Many tryptic peptide sequences are conserved over different bacterial taxa and are therefore not-informative to describe the taxonomic diversity or functional properties of the sample. Askenazi, Marto, and Linial (2010) developed the Pep2Pro web service to identify taxon-specific peptides. However, they used a restricted definition of taxon-specificity by only retaining peptides unique to a single taxon as defined in the NCBI taxonomy. In this paper, we present Unipept (http://unipept.ugent.be), a web application that supports biodiversity analysis of large and complex metaproteome samples using tryptic peptide information obtained from shotgun MS/MS experiments. Its underlying index structure is designed to quickly retrieve all occurrences of a tryptic peptide in UniProt entries. Taxon-specificity of the tryptic peptide is successively derived from these occurrences using a novel lowest common ancestor approach that is robust against taxonomic misarrangements, misidentifications, and inaccuracies. Not taking into account this identification noise would otherwise result in drastic loss of information. Dynamic treemaps visualize the biodiversity of metaproteome samples, which eases the exploration of samples with highly complex compositions. The potential of Unipept to gain novel insights into the biodiversity of a sample is evaluated by reanalyzing publicly available environmental metaproteome data sets from phyllosphere bacteria (Delmotte et al., 2009) and the human gut (Verberkmoes et al., 2009).

## 2.1.2 Materials and methods

### *2.1.2.1 Database construction*

Unipept uses a MySQL database as its storage backbone. The database is populated using a custom data processing pipeline written in the Java programming language. This pipeline integrates the NCBI Taxonomy Database (Wheeler et al., 2004) with the UniProt Knowledgebase (UniProtKB; Wu et al. (2006)). The complete NCBI Taxonomy Database is fetched from the NCBI FTP server. While processing the downloaded database, only those records with a name class containing "scientific name" are retained, accounting for about 882 000 records. Unipept mainly relies on the hierarchical structure of the NCBI taxonomy and the rank information it assigns to the different nodes. Careful use of indexes provides fast access to the necessary taxonomic information. As a second data source, Unipept uses the XML version of the UniProt Knowledgebase. UniProtKB consists of two parts, Swiss-Prot and TrEMBL, containing more than 17 million protein entries, including proteins from complete and reference proteomes. The Java pipeline iterates over all protein entries and performs an *in silico* trypsin digest on every protein. The individual peptides are added to the Unipept database, together with additional metadata from the UniProt entry such as organism information and various cross references. Using UniProt release 2012_07, this establishes a comprehensive catalogue of over 250 million tryptic peptides.

The size of the UniProt database is discussed in more detail in Section 5.2.10.

Because Unipept was developed to work with mass spectrometry data as input, two additional extensions to the data integration pipeline were made. First, peptides with a length smaller than five or greater than fifty amino acids are discarded. This is done because most mass spectrometers have a limited mass range and small peptides are difficult to distinguish by mass alone (Eidhammer et al., 2007). In addition, a lower limit of five amino acids has no real impact on biodiversity analysis, since shorter peptides are generally found in the genome of organisms across all kingdoms of life. The second extension takes into account the difficulties to discern the isobaric amino acids isoleucine (I) and leucine (L). Therefore, a duplicate version of every tryptic peptide is stored in which the difference between I and L is ignored. When running queries on Unipept, users can decide whether differentiating between I and L is relevant or not.

#### *2.1.2.2 Web application*

To allow ubiquitous access to the Unipept database, a web based front-end was developed using Ruby on Rails (RoR), which is accessible at http://unipept.ugent.be. RoR is a web application framework for the Ruby programming language, which allows rapid prototyping and has built-in support for the creation of RESTful web services. The application consists of two basic functionalities: a peptide-based taxonomic identification and a multi-peptide dynamic diversity treemap analysis.

#### *2.1.2.3 Single peptide analysis*

Single peptide analysis was since renamed to tryptic peptide analysis.

With single peptide analysis, users submit a single tryptic peptide with potential missed cleavages. Individual tryptic peptides can be 5 to 50 residues long. The application responds with a list of all UniProt entries wherein the peptide was found. For every UniProt entry in this result set, the complete taxonomic lineage is derived from the NCBI taxonomy. These lineages are successively used to compute the common lineage of the peptide as the lowest common ancestor (LCA) of the organisms. The resulting information is presented as a comprehensible table that contains all matched UniProt entries and visualized using an interactive JavaScript tree view that bundles all taxonomic lineages. The peptide analysis page displayed by Unipept also contains a hyperlink that allows to directly BLAST the peptide against a series of NCBI hosted databases.

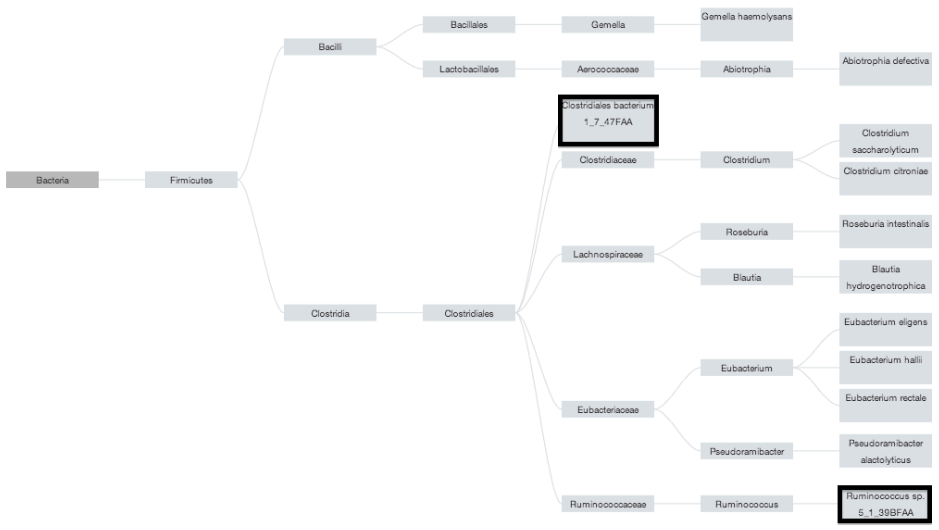
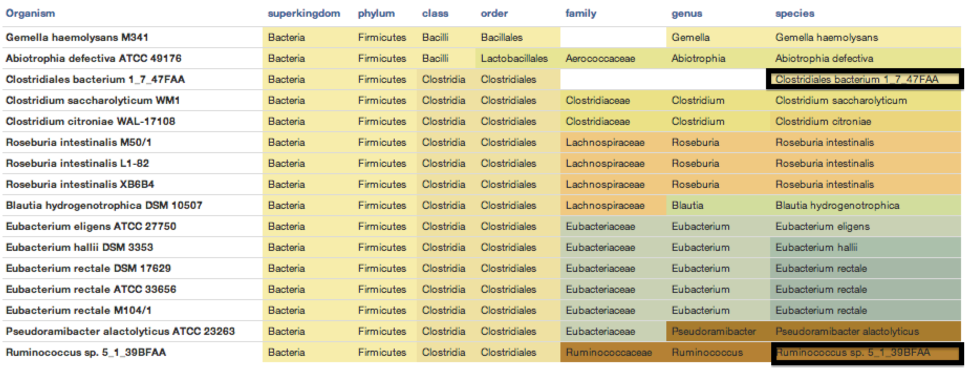


**Figure 2.1** Tree view bundling the complete taxonomic lineages of all UniProt entries whose protein sequence contains the tryptic peptide ELTSLVDGPLSGEVK. Figure shows taxonomic lineages before cleanup procedures are in effect, resulting in invalid species nodes (marked with thick border) *Clostridiales bacterium 1_7_47FAA* (NCBI taxon ID 457421) and *Ruminococcus* sp. *5_1_39BFAA* (NCBI taxon ID 457412) showing up in the tree. The phylum Firmicutes is the lowest common ancestor (LCA) of all taxonomic lineages taken from the UniProt entries in which the peptide was found. A dynamic version of this tree view is included in the web application, as the number of levels and nodes bundled in a tree might grow large for peptides that are universally found over multiple different lineages. Take into account that due to gaps in the hierarchical structure of the NCBI taxonomy, taxonomic levels are not always properly aligned in the tree view.



| Organism | superkingdom | phylum | class | order | family | genus | species |
|---|---|---|---|---|---|---|---|
| Gemella haemolysans M341 | Bacteria | Firmicutes | Bacilli | Bacillales |  | Gemella | Gemella haemolysans |
| Abiotrophia defectiva ATCC 49176 | Bacteria | Firmicutes | Bacilli | Lactobacillales | Aerococcaceae | Abiotrophia | Abiotrophia defectiva |
| Clostridiales bacterium 1_7_47FAA | Bacteria | Firmicutes | Clostridia | Clostridiales |  |  | Clostridiales bacterium 1_7_47FAA |
| Clostridium saccharolyticum WM1 | Bacteria | Firmicutes | Clostridia | Clostridiales | Clostridiaceae | Clostridium | Clostridium saccharolyticum |
| Clostridium citroniae WAL-17108 | Bacteria | Firmicutes | Clostridia | Clostridiales | Clostridiaceae | Clostridium | Clostridium citroniae |
| Roseburia intestinalis M50/1 | Bacteria | Firmicutes | Clostridia | Clostridiales | Lachnospiraceae | Roseburia | Roseburia intestinalis |
| Roseburia intestinalis L1-82 | Bacteria | Firmicutes | Clostridia | Clostridiales | Lachnospiraceae | Roseburia | Roseburia intestinalis |
| Roseburia intestinalis XB6B4 | Bacteria | Firmicutes | Clostridia | Clostridiales | Lachnospiraceae | Roseburia | Roseburia intestinalis |
| Blautia hydrogenotrophica DSM 10507 | Bacteria | Firmicutes | Clostridia | Clostridiales | Lachnospiraceae | Blautia | Blautia hydrogenotrophica |
| Eubacterium eligens ATCC 27750 | Bacteria | Firmicutes | Clostridia | Clostridiales | Eubacteriaceae | Eubacterium | Eubacterium eligens |
| Eubacterium hallii DSM 3353 | Bacteria | Firmicutes | Clostridia | Clostridiales | Eubacteriaceae | Eubacterium | Eubacterium hallii |
| Eubacterium rectale DSM 17629 | Bacteria | Firmicutes | Clostridia | Clostridiales | Eubacteriaceae | Eubacterium | Eubacterium rectale |
| Eubacterium rectale ATCC 33656 | Bacteria | Firmicutes | Clostridia | Clostridiales | Eubacteriaceae | Eubacterium | Eubacterium rectale |
| Eubacterium rectale M104/1 | Bacteria | Firmicutes | Clostridia | Clostridiales | Eubacteriaceae | Eubacterium | Eubacterium rectale |
| Pseudoramibacter alactolyticus ATCC 23263 | Bacteria | Firmicutes | Clostridia | Clostridiales | Eubacteriaceae | Pseudoramibacter | Pseudoramibacter alactolyticus |
| Ruminococcus sp. 5_1_39BFAA | Bacteria | Firmicutes | Clostridia | Clostridiales | Ruminococcaceae | Ruminococcus | Ruminococcus sp. 5_1_39BFAA |

**Figure 2.2** Table showing complete taxonomic lineages of all UniProt entries whose protein sequence contains the tryptic peptide ELTSLVDG-PLSGEVK. Each row represents the complete taxonomic lineage of a single UniProt entry. First column contains the name extracted from the UniProt entry, followed by columns representing valid taxonomic ranks ordered from superkingdom on the left to forma on the right. Only ranks used in the complete lineages of the entries in which the peptide was found are displayed as columns in the table. To ease the interpretation of the table, rows are ordered according to the taxonomic names in the taxonomic rank columns (evaluated from left to right) and cells containing the same taxonomic name are marked with the same color code per column. The table shows taxonomic lineages before cleanup procedures are in effect, resulting in invalid species nodes (marked with thick border) *Clostridium bacterium 1_7_47FAA* (NCBI taxon ID 457421) and *Ruminococcus* sp. *5_1_39BFAA* (NCBI taxon ID 457412) showing up in the table.

When single peptide analysis is applied, for example, to the tryptic peptide ELTSLVDGPLSGEVK (http://unipept.ugent.be /sequences/ELTSLVDGPLSGEVK/equateIL), it results in the lineage tree depicted in Figure 2.1 and the lineage table shown in Figure 2.2. These results show that the peptide was found in 17 UniProt entries, covering 12 different species (10 after taxonomic cleanup; see below) and 8 different genera. In particular, the peptide was found twice in the *Clostridium saccharolyticum WM1* genome, in both copies of the duplicated tal2 gene (UniProt entries D9R9N2 and D9R6P3). Note that not all UniProt entries from the result set were identified up to the species level.

All UniProt entries in which the peptide was found have a common phylum (Firmicutes), which was computed as the LCA of taxonomic identifications mentioned in the UniProt entries.

**Table 2.1** Overview of the benchmark datasets used. The first dataset consists of two samples of the human gut microbiota of twins from a study by Verberkmoes et al. (2009). The second dataset includes six samples of aerial plant surfaces of soybean (*Glycine max*), clover (*Trifolium repens*) and *Arabidopsis thaliana* from a study by Delmotte et al. (2009). The first two columns contain the name of the dataset and their origin followed by columns containing the number of peptides included in the dataset, the number of peptides after filtering, the number of peptides after filtering duplicates when equating isoleucine and leucine and the number of peptides found by Unipept while filtering duplicates and equating isoleucine and leucine.

| NAME | SAMPLED ENVIRONMENT | # PEPTIDES | # PEPTIDES AFTER DEDUPLICATION | # PEPTIDES AFTER DEDUPLICATION AND I=L | # UNIPEPT MATCHES |
|---|---|---:|---:|---:|---:|
| sample 7 | human gut | 3 895 | 1 854 | 1 809 | 1 771 |
| sample 8 | human gut | 5 447 | 2 704 | 2 644 | 2 555 |
| A. thaliana | phyllosphere | 10 019 | 2 930 | 2 914 | 2 397 |
| Clover 1a | phyllosphere | 8 418 | 2 913 | 2 901 | 2 264 |
| Clover 1b | phyllosphere | 9 636 | 2 591 | 2 581 | 1 935 |
| Clover 2 | phyllosphere | 1 862 | 645 | 643 | 600 |
| Soybean 1 | phyllosphere | 15 140 | 5 151 | 5 134 | 4 758 |
| Soybean 2 | phyllosphere | 19 493 | 3 583 | 3 572 | 2 740 |

The LCA of a set of taxonomic identifications is computed by constructing a lineage table in which the rows represent the complete taxonomic lineage of a single UniProt entry. The columns represent valid taxonomic ranks, ordered from superkingdom on the left to forma on the right. Subsequently, the columns are scanned from left to right and the rightmost column where every row has the same value is called the LCA. Figure 2.2 shows the lineage table for the example tryptic peptide, in which the phylum column clearly is the rightmost column containing equal values. This naive approach of scanning the lineage table suffers from two major problems, both of which have been addressed in the application.

A first confounding factor is that some intermediate taxonomic levels might be missing in the complete lineage of a taxon in the NCBI taxonomic hierarchy, resulting in gaps within some of the columns of the lineage table. This is exemplified by the complete lineage of the species *Gemella haemolysans* (Figure 2.2), which was not assigned to a valid family. If a gap would be treated as a separate value in the LCA scan, poor identification results would be obtained for LCAs with the rank of species and genus where a better classification is often possible. If instead gaps would be completely ignored, problems would arise at higher taxonomic levels in cases where only some of the lineages have values for a certain level. A hybrid solution was chosen in which gaps are treated as separate values, except at the species and the genus levels. The outcome of this heuristic approach is in line with what could be intuitively expected from an LCA algorithm. In effect, taxonomic lineages of UniProt entries that have not been identified up to the species or genus levels are discarded if the peptide occurs in another UniProt entry that has an identification that is more specific. This avoids that very specific identifications (e.g., at the species level) in UniProt would be masked by less specific identifications (e.g., at the family level) in determining the common lineage for a given peptide.

A second and more substantial problem impeding accurate identification of the common lineage of a given peptide is the quality of the NCBI taxonomy on the one hand, and inaccurate or incorrect taxonomic identifications in UniProt entries on the other hand. This combination of taxonomic misarrangements, misidentifications, and inaccuracies can result in drastic loss of information. If a single peptide occurs in 6 bacteria identified as belonging to the same species and 1 organism classified as *Unidentified bacteria* (NCBI taxon ID 77133; rank species), the LCA of the peptide would be set to Bacteria (rank superkingdom) instead of the more fine-grained species name. To counter this problem, taxonomic nodes that occur in the NCBI taxonomy but have no official taxonomic status are heuristically invalidated based on a set of regular expressions. For example, taxonomic nodes containing words like "uncultured", "unspecified" or "undetermined" in their name are marked as invalid in the database. In addition, species nodes with names containing a number are invalidated. As a result of this procedure, taxonomic nodes like *Uncultured cyanobacterium* (NCBI taxon ID 1211), *Environmental samples* (NCBI taxon ID 1166701) and *Ruminococcus* sp. *5_1_39BFAA* (NCBI taxon ID 457412) are ignored while calculating the LCA, but their higher taxonomic nodes are retained in the procedure if they are not invalidated themselves. This results in the invalidation of about 382 000 of the 882 000 taxonomic nodes (42%) in the Unipept database. Almost all of these invalidations (374 000 or 98%) occurred at the species level, reducing the number of valid species nodes to 314 000 or 46% of the original amount. The thick boxes in Figure 2.1 and Figure 2.2 show the effect of this taxonomic cleanup for the tryptic peptide ELTSLVDGPLSGEVK that was already used as an example above. The cleanup procedure will remove the artificial species nodes labeled *Clostridium bacterium 1_7_47FAA* (NCBI taxon ID 457421) and *Ruminococcus* sp. *5_1_39BFAA* (NCBI taxon ID 457412), both from the tree view and the lineage table.

This blacklist was later expanded to include names like "metagenome", "sample", and "library".

The implementation of the advanced missed cleavage handling is covered in Section 5.2.3.

As another improvement in the identification pipeline, single peptide analysis of Unipept also deals with missed cleavages in the peptides fed to the pipeline by intersecting the result sets of the UniProt entries in which the individual tryptic peptides are found. By doing so, miscleavage support builds upon the same tryptic peptide index of the Unipept database with some additional post processing. This incurs a minimal performance overhead and no additional memory requirements. For example, if an *in silico* trypsin digest is performed on the peptide ELTSLVDGPLSGEVKATTTDAEGMLAEGR, the resulting two subpeptides ELTSLVDGPLSGEVK and ATTTDAEGMLAEGR respectively have an LCA of Firmicutes (phylum) and Clostridiales (order). The single peptide analysis pipeline of Unipept however intersects the result sets of the subpeptides before computing the LCA, and identifies the original peptide as specific to the species *Pseudoramibacter alactolyticus*. This is an accuracy improvement of several taxonomic ranks, and is in line with the result one would come up with after a more time consuming BLAST search. Because of the way missed cleavage support is implemented, Unipept incurs no limitation on the number of missed cleavages provided that at least one tryptic subpeptide has a minimal length of five.

#### *2.1.2.4 Multi-peptide analysis*

Multi-peptide analysis was since renamed to metaproteomics analysis.

Unipept contains features to help analyze lists of tryptic peptides, e.g., extracted from an environmental sample. These features build on the algorithms of single peptide analysis as discussed in the previous section. Because missed cleavage support has a negative impact on the performance of the analysis due to the fact that only the LCAs of tryptic peptides are cached, users have the option to perform an alternative treatment of miscleaved peptides. In the alternative approach, an additional preprocessing step takes place during which an *in silico* trypsin digest is performed on every peptide fed to the pipeline. This trypsin digest splits peptides with missed cleavages and computes the LCA of each individual subpeptide. This alternative is a faster but less accurate approach to deal with missed cleavages. Apart from choosing which approach is used to handle missed cleavages, users also have the option to filter out duplicate peptides or to equate isoleucine and leucine.

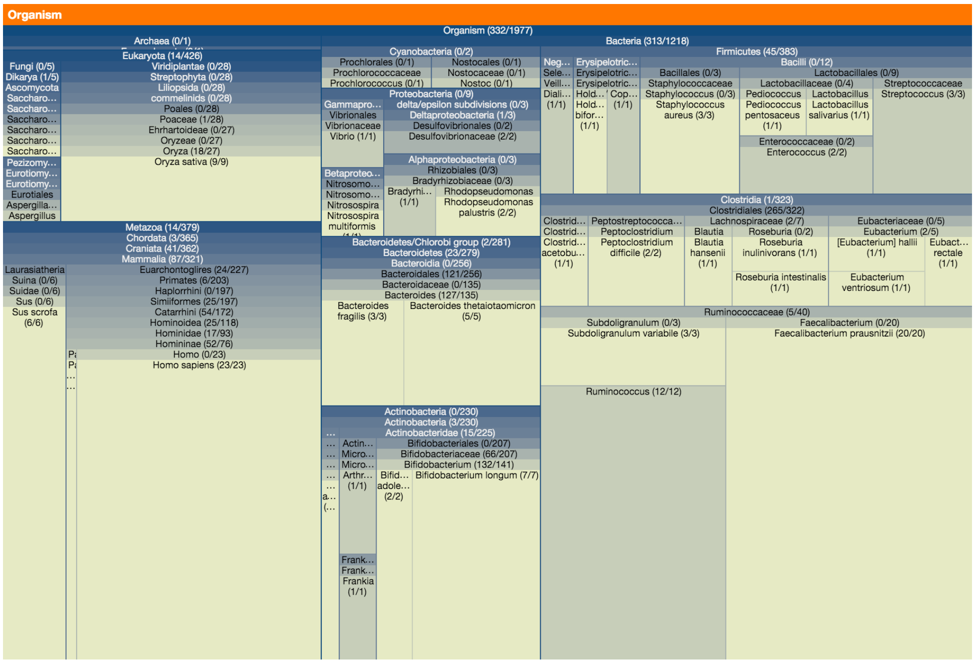


**Figure 2.3** Treemap visualizing bacterial diversity in human gut metaproteome sample 7 as determined by Verberkmoes et al. (2009). Multi-peptide analysis performed with deduplicated peptides and equating isoleucine (I) and leucine (L). Unipept displays a dynamic version of this treemap that allows zooming in and out to areas of interest, in order to gain further insight in the complex composition of the diversity in the sample.

In a consecutive step of the processing pipeline, the LCA of every tryptic peptide that was submitted is calculated as described in the section on single peptide analysis. These LCAs are then bundled internally into a frequency table and visualized using an interactive treemap (Figure 2.3). This treemap intuitively combines a multilayer histogram-like graphical representation with hierarchical data. Every square in the treemap corresponds to a taxonomic node in the NCBI taxonomy. The size of the square is proportional to the number of peptides having that taxonomic node as their LCA. Squares are tiled hierarchically according to their occurrence in the taxonomic lineages. Color codes correspond to the different taxonomic ranks. This way, users can see at a glance which organisms are present in a metaproteome sample and to what extent. To allow better exploration of the diversity of complex samples, users can dynamically zoom in to an area of interest by clicking on a node (e.g., Bacteria or Clostridiales) or zoom out by right clicking the treemap. As a result, the treemap is restricted to the list of peptides from the sample that are specific to the level that is displayed. To further aid the exploration of the biodiversity composition of the sample, hovering the mouse over a taxonomic node pops up a pie chart showing a quantification of the occurrences of the child nodes in the sample. All taxa found as LCAs of peptides in the sample are also shown in a hierarchical outline that follows the NCBI taxonomy. This hierarchical outline is dynamically updated as users navigate the treemap above it. By clicking on a taxonomic node in the hierarchy, two lists of sample peptides associated with that taxon are shown: a list of peptides that have the taxon as their LCA, and a list of peptides whose LCA is the taxon or one of its descendants in the NCBI taxonomy. The number of peptides in both lists also appear in between brackets after each taxon name in the treemap and the hierarchical outline.

Below the treemap display, Unipept lists all tryptic peptides from the sample that were not found in any protein sequence in UniProt. These mismatched peptides are provided with a hyperlink to BLAST them against a series of NCBI hosted databases.

The multi-peptide analysis results can be exported to a Microsoft Excel compatible CSV file. For their convenience, users can also assign names to searches in order to easily recognize results when running multiple analyses in different browser windows.

### 2.1.2.5 Hardware and performance

The current server is dual-CPU 2.6 GHz Intel Xeon octacore processor with 128 GB of memory.

The time to construct the database from scratch was eventually reduced to just 7 hours.

The LCA of every peptide is now automatically computed while constructing the database.

Unipept runs on a virtual Debian machine with a dedicated 2.6 GHz Intel Xeon hexacore processor and 24 GB of memory. The database uses MySQL 5.5 as a database management system in combination with the InnoDB storage engine. On this machine, the construction and indexing of the entire database takes about one week. Because of this long construction time, the implementation of a smart update mechanism is planned. Updating the database instead of building it up from scratch every time will allow to incorporate monthly UniProt updates without considerable downtime. The total size of the resulting database is 82 GB with 39 GB used for data and 43 GB used for indexes. To improve performance of the application, a lot of time was spent on optimizing the database schema and selecting the right indexes. An even higher level of performance could be achieved by increasing the server memory so the full database, or at least the index, fits into memory. The bottleneck of the web application is the computation of the LCA of every peptide searched during multi-peptide analysis. To address this bottleneck, the LCA of a peptide is only computed once, the first time it is needed. After calculation, the LCA is cached into the database so it can easily be retrieved the next time without the need for recalculation.

### *2.1.2.6 Benchmark datasets*

To evaluate the potential of Unipept to gain novel insights into the biodiversity of a sample, two publicly available datasets were reanalyzed (Table 2.1). The first dataset consists of two samples of the human gut microbiota of twins from a study by (Verberkmoes et al., 2009). Of these two samples, 97.6% and 96.5% of the tryptic peptides could be matched by Unipept. The second dataset includes six samples of aerial plant surfaces of soybean (*Glycine max*), clover (*Trifolium repens*) and *Arabidopsis thaliana* from a study by Delmotte et al. (2009). Of these phyllosphere samples, between 75% and 93% of the tryptic peptides could be matched by Unipept. The benchmark datasets are readily available from the Unipept website and can be directly analyzed from there.

## 2.1.3 Results and discussion

### *2.1.3.1 Accuracy of peptide-based biodiversity analysis*

Due to the more conserved nature of protein sequences, peptide-based identification is generally more reliable than DNA-based identification. For the same reason, tryptic peptides can be matched against a reference database using exact string matching algorithms. These are much faster than the inexact string matching algorithms that are commonly used to match DNA sequences against a reference database and more accurate than most heuristic implementations for inexact string matching (like BLAST).

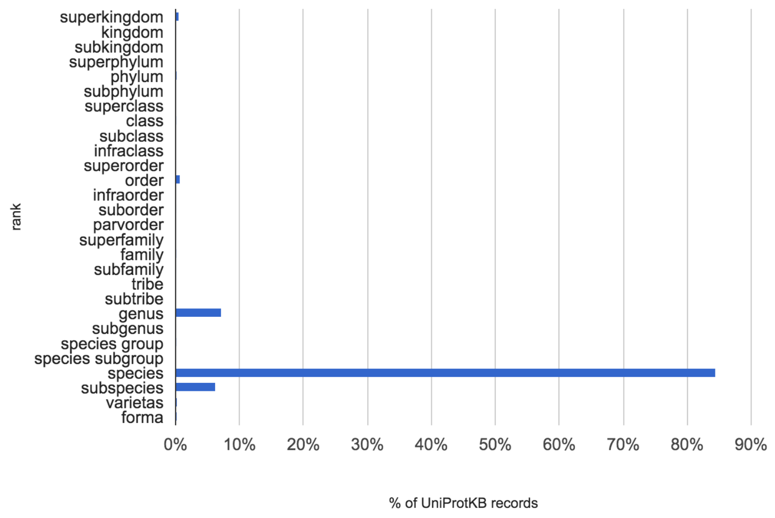


**Figure 2.4** Identification depth of UniProt entries after cleanup of NCBI taxonomy.

However, other crucial factors that affect both peptide-based and DNA-based identification are coverage and quality of the diversity information in the reference database. In case of the Unipept database, accuracy of the identification strongly depends on the reliability of the hierarchical structure of the NCBI taxonomy and the correctness and accuracy of the identifications included in the UniProt entries. However, many UniProt entries are linked to nodes in the NCBI taxonomy that either have no rank or have no valid status. Both cases often correspond to artificial taxonomic nodes that in theory should not be included in the taxonomy, but are merely added as placeholders upon submission in case no valid taxa are appropriate or to specify organism-level identification. The Unipept analysis pipeline therefore contains noise filtering procedures to safeguard sound biodiversity analysis of metaproteome sample data. These include cleaning the NCBI taxonomy by rendering some taxonomic nodes invalid, ignoring nodes that are invalid or have no rank and masking less accurate identifications when covered by more accurate identifications during the LCA scanning procedure. After cleanup of the NCBI taxonomy and taking the first valid rank in the lineage to which the UniProt entries were assigned (ignoring lineage nodes that are invalid or have no rank), 84.42% of the UniProt entries are identified at the species level, 7.19% at the genus level, 6.25% at subspecies level, 0.83% at the order level and 0.57% at the superkingdom level (Figure 2.4; occurrences of all other levels fall below 0.5%).

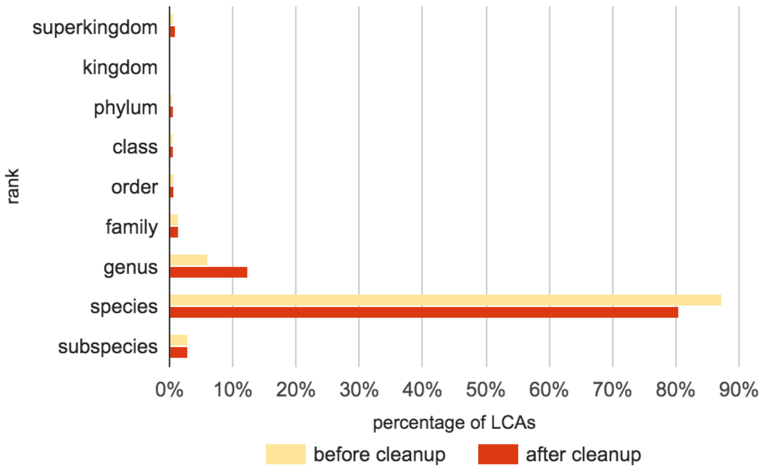


**Figure 2.5** Distribution of LCA taxonomic ranks computed for all tryptic peptides extracted from UniProt entries before (top) and after (bottom) taxonomic cleanup. Histogram restricted to most frequently observed LCA taxonomic ranks.

The individual effects of the three noise filtering procedures interfere with one another during the computation of the LCAs of the peptides. The impact of noise filtering is therefore most easily seen from individual cases as exemplified in the materials and methods section. However, in order to get a global impression of the impact of noise filtering, we can observe the shifts in the taxonomic ranks of the LCAs as they are computed for all tryptic peptides in the Unipept database before and after cleanup. A histogram of this taxonomic rank distribution is shown in Figure 2.5. The most striking observation is that over 80% of the tryptic peptides are seemingly species-specific, which would confirm the hypothesis that tryptic peptides can be used to make accurate identifications at the species level. Making a final claim on this, however, is too strong as the observation is somewhat blurred by an under-sampling of biodiversity in the UniProt database. Many known taxa have no or a low number of organisms being sequenced in UniProt, and for those organisms that have been sequenced usually only a small fraction of the proteome is available. We can only observe that a large fraction of the peptides only occurs in one or a few protein sequences in UniProt, without confirmation that they do not occur in organisms that have not been (completely) sequenced.

A second notable observation from Figure 2.5 is the 6 percentage points drop in species-specific identifications after NCBI taxonomy cleanup and an equivalent rise in genus-specific identifications. This might seem counterintuitive at first, as the main goal of cleaning up was to achieve better identifications. However, better should be interpreted as more realistic in this case, not necessarily as more accurate in the sense of identifications at more specific taxonomic ranks. A twofold explanation of this species/genus shift can be made. 98% of the taxonomic nodes from the NCBI taxonomy that have been invalidated are artificial nodes at the species level. Combined with the observation that most peptides only occur in a single UniProt entry, this results in an LCA at the genus level instead of at the (invalidated) species level. It requires peptides to occur in multiple UniProt entries for one of the intended effects of taxonomic cleanup, namely more fine-grained identifications, to possibly show some effect. As peptides that have more abundant occurrences in UniProt are far less frequent in the complete Unipept database, the shift of LCA identifications from less accurate to more accurate taxonomic ranks is occluded in the histogram of Figure 2.5. However, the effect becomes much more prominent if we observe the shift in LCA taxonomic ranks before and after cleanup for peptides identified from the human gut metaproteome sample 7 as determined by Verberkmoes et al. (2009). This is explained by the fact that peptides identified from an environmental sample are biased towards peptides that are more abundant in the reference database. It also underscores the importance of filtering the identification noise during the biodiversity analysis of metaproteome samples.

#### *2.1.3.2 Comparison with Pep2Pro*

The Pep2Pro application has since been deprecated in favor of the PIR Peptide Match Service (Chen et al., 2013).

The complete peptide dictionary Pep2Pro (Askenazi, Marto, and Linial, 2010) offers several web services for metaproteome analysis that somewhat resemble features offered by Unipept. Both systems precompute peptide indexes from UniProt, where Pep2Pro uses a restricted index built from clusters in the UniRef100 database (in essence, a deduplicated version of UniProtKB). Each system also builds separate indexes that either consider leucine and isoleucine as distinct or equal residues. However, there are some important differences between both systems that have considerable implications on the results of the proteome analysis.

Pep2Pro uses a hash table to index all 6-mers extracted from UniRef100 protein sequences. This provides flexibility to find exact matches for all peptides, but requires multiple steps to exactly match peptides from their decomposed 6-mers. On the other hand, Unipept restrictively extracts tryptic peptides having 5 to 50 residues from UniProt entries, to build an index that can find exact matches in a single step. As a result, single peptide matching is considerably faster in Unipept compared to Pep2Pro, but is restricted to tryptic peptides only. The index construction strategy of Unipept is motivated by the observation that trypsin is the most commonly used protease in metaproteome studies. As such, flexibility was traded for a dramatic increase in performance, which results in a much faster response time.

Unipept takes a completely different angle in handling taxon-specificity when performing peptide-based biodiversity analysis, compared to Pep2Pro. The metaproteome analysis tool offered by Pep2Pro only retains peptides that are matched with UniRef entries having the same NCBI taxon identifier, without making any distinction between the taxonomic ranks of these NCBI taxa. Usually, these so-called unique peptides only form a minor fraction of most peptides identified from metaproteome samples (between 4.1% and 14.1% in the Verberkmoes and Delmotte data sets). As such, a majority of the information content gets lost in the biodiversity analysis.

The tryptic peptide EPGSLGEPLYLDVATTLR is considered non-unique by Pep2Pro as it was found in one protein sequence of the bacterial species *Eubacterium eligens* and three protein sequences of the bacterial species *E. rectale*. It is therefore ignored in the biodiversity analysis. Note that even if this peptide was not found in the *E. eligens* protein, Pep2Pro would still not have considered it to be specific for the species *E. rectale*. This due to the fact that the three corresponding *E. rectale* UniProt entries are assigned a separate NCBI taxon identifier without a taxonomic rank, referring to the individual bacterial strains that were sequenced. Considering these artificial NCBI records as separate taxa really makes no sense. The observation that the tryptic peptide EPGSLGEPLYLDVATTLR is specific to the genus *Eubacterium* is still considered valuable information by Unipept. Taxon-specificity is therefore computed using a cleaned up version of the NCBI taxonomy as a guidance hierarchy during the LCA scanning procedure. As a result, Unipept uses the most granular information it can derive from peptide matches during its biodiversity analysis. This does not necessarily need to be restricted to species-specificity, as it still gives valuable insights if the taxonomic information derived from all tryptic peptides identified from a metaproteome sample are bundled into a treemap visualization.

#### *2.1.3.3 Microbiological evaluation*

Reanalysis of sample 7 from the Verberkmoes et al. (2009) study yielded 1 292 bacteria-specific peptides among which peptides from Bifidobacteriaceae (224 specific peptides of which 39 could be assigned to *B. adolescentis* and 11 to *B. longum*), Bacteroides (162 specific peptides of which 3 could be assigned to *B. fragilis* and 9 to *B. thetaiotaomicron*) and Clostridiales (377 specific peptides of which 79 could be assigned to the Ruminococcaceae family with *Ruminococcus* [42 specific peptides] and *Faecalibacterium* [30 specific peptides] as predominant genera) were most prevalent. Similarly, reanalysis of sample 8 from the Verberkmoes et al. (2009) study yielded 2 114 bacteria-specific peptides among which again peptides from Bifidobacteriaceae, Bacteroides, and Clostridiales were predominant. Compared to the application of the Pep2Pro software (Askenazi, Marto, and Linial, 2010), this provides us with a much larger list of species-specific peptides combined with peptides that are specific for higher taxonomic ranks, which corresponds more accurately to the expected bacterial diversity in the human gut microbiome. It also confirms that members of the Bacteroides, Bifidobacterium, and Clostridiales are metabolically dominant in the human gut ecosystem (Verberkmoes et al., 2009), allows to assign enzymes and thus metabolic functions to individual microbial species and reveals shared metabolic potential in higher taxonomic categories. It also confirms that the microbiome of the subject from which sample 8 was taken comprised a very high number of peptides specific to the species *Eubacterium rectale* (Askenazi, Marto, and Linial, 2010).

#### *2.1.3.4 Future developments*

The Unipept web application presented here is a novel approach to peptide-based biodiversity analysis. The LCA algorithm has the significant advantage, when compared to the Pep2Pro tool, of using noise filtering algorithms to provide the most accurate results. Restricting the scope of the application to tryptic peptides also allows for a highly optimized index yielding excellent performance. To the best of our knowledge, no index structures exist that are both sufficiently fast and memory efficient to allow exact substring matching of generic peptides over all UniProt entries. Further research is planned to have Unipept support proteases other than trypsin, without dramatic loss of performance. In addition to being fast, the application is also easy to use and presents the results in a visually appealing way. Now that the basic infrastructure is put in place, additional support for the comparison of multiple samples is planned. Further improvements will also include support for functional analysis of metaproteome samples by adding the functional annotations of the Gene Ontology project (Ashburner et al., 2000) to the Unipept database.

## 2.2 The Unipept metaproteomics analysis pipeline

This section contains a verbatim copy of the application note by Mesuere et al. (2015) as published in Proteomics.

**Abstract** — Unipept, available at http://unipept.ugent.be, is a web application that offers a user friendly way to explore the biodiversity of complex metaproteome samples by providing interactive visualizations. In this article, the updates and changes to Unipept since its initial release are presented. This includes the addition of interactive sunburst and treeview visualizations to the multi-peptide analysis, the foundations of an API and a command line interface, updated data sources and the open-sourcing of the entire application under the MIT license.

### 2.2.1 Introduction

Unipept (http://unipept.ugent.be) integrates a fast index of tryptic peptides built from UniProt Knowledgebase (UniProtKB; The UniProt Consortium (2014)) entries with cleaned up information from the NCBI Taxonomy Database (Federhen, 2012) to allow for biodiversity analysis of metaproteome samples. Users can submit tryptic peptides obtained from shotgun MS/MS experiments to which the application responds with a list of all UniProt entries containing that peptide. The NCBI Taxonomy Database is used to compute the complete taxonomic lineage of every UniProt entry in the result set. Subsequently, these lineages are combined to compute the common lineage of the submitted peptide. Of this common lineage, the most specific taxonomic node is determined as the lowest common ancestor (LCA) using a robust LCA scanning algorithm. The resulting information is visualized using an interactive JavaScript treeview that bundles all taxonomic lineages, accompanied by a comprehensible table that contains all matched UniProt entries.

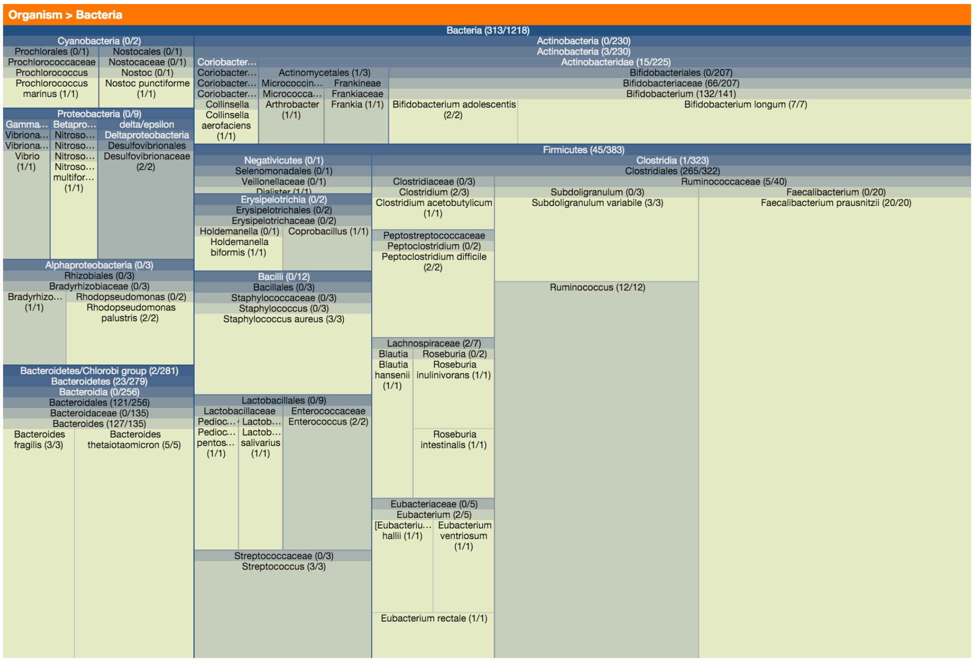


**Figure 2.6** The treemap visualization available in Unipept when running a multi-peptide analysis on sample 7 as determined by Verberkmoes et al. (2009), zoomed in on Bacteria.

Users can also submit a list of tryptic peptides. In this case, the LCA is calculated for every submitted peptide as described above. These LCAs are then bundled into a frequency table and visualized on the results page using an interactive treemap (Figure 2.6). This treemap displays hierarchical data in a multilayer histogram-like graphical representation. The squares in the treemap each correspond to a taxonomic node in the NCBI taxonomy, with their size proportional to the number of peptides having that taxonomic node as their LCA. The cleaned up hierarchy of the NCBI Taxonomy is used to tile the squares according to their occurrence in the taxonomic lineages. These squares are color coded according to their taxonomic ranks. This graphical representation allows users to see at a glance which organisms are present in a metaproteome sample and to what extent. The treemap is interactive and can be manipulated by clicking on individual nodes. This makes it possible for users to zoom in to an area of interest (e.g., Bacteria or Firmicutes).

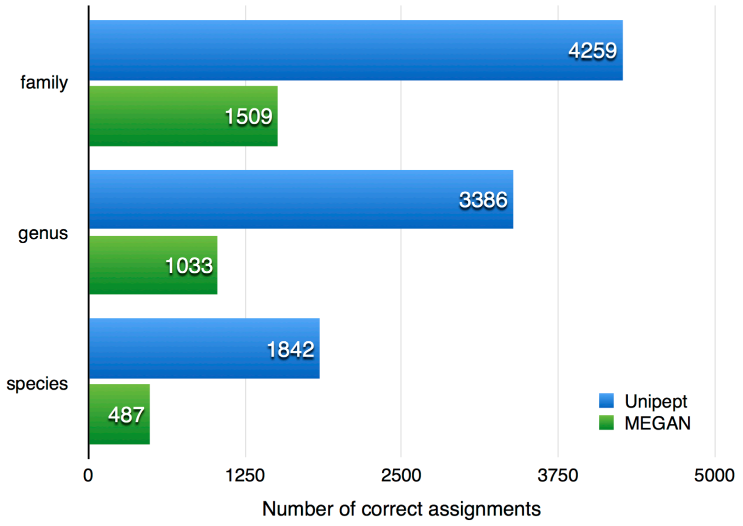


**Figure 2.7** Bar chart showing experimental data from Tanca et al. (2013) with the number of correct taxonomic assignments at the family, genus, and species level from the lab assembled 9MM sample using the NCBI-BFV database. The amount of incorrect identifications are 6%, 10%, and 15% respectively for Unipept, and 9%, 18%, and 34% for MEGAN.

Since its release in 2012 (Mesuere et al., 2012), the Unipept application was praised for its ease of use and great data visualizations (Tanca et al., 2013; Kolmeder and Vos, 2014; Seifert et al., 2013). The Unipept LCA approach was compared with the commonly used taxonomic analysis in MEGAN (Huson et al., 2011) by Tanca et al. (2013). Unipept consistently outperformed MEGAN by a factor of three on both the family, genus, and species level with only half the number of misassignments (Figure 2.7). In this article we present some of the improvements and new features of Unipept since its original publication.

## 2.2.2 What's new

### Sunburst

With complex samples containing a diverse range of taxa, the treemap (Figure 2.6) representation quickly becomes cluttered which makes it hard to get a clear insight into the results. To resolve this problem, a new sunburst visualization (Andrews and Heidegger, 1998) was built into Unipept (Figure 2.8) using the D3.js framework (Bostock, Ogievetsky, and Heer, 2011). The sunburst diagram displays the same data as the treemap, but as an interactive multi-level pie chart. The center of this pie chart represents the root node with several concentric rings around it. These rings are divided into slices representing the child nodes in the taxonomic hierarchy of the aligning more central slice. The size of the slices corresponds to the number of peptides having an LCA equal to that taxonomic node or any of its children.

In a later version, breadcrumbs were added to allow users to keep track of their position in the phylogenetic tree.

The sunburst diagram provides a more comprehensive view by displaying only four levels at a time. Users can see more levels by clicking on a slice of interest. The node that was clicked then becomes the center of the sunburst and the four levels below it are displayed. By clicking on the center of the sunburst, users can zoom out one level. By hovering the mouse over a slice, a tooltip is displayed with more information about the taxonomic node associated with that slice and the number of peptides belonging to that taxonomic node or any of its children.

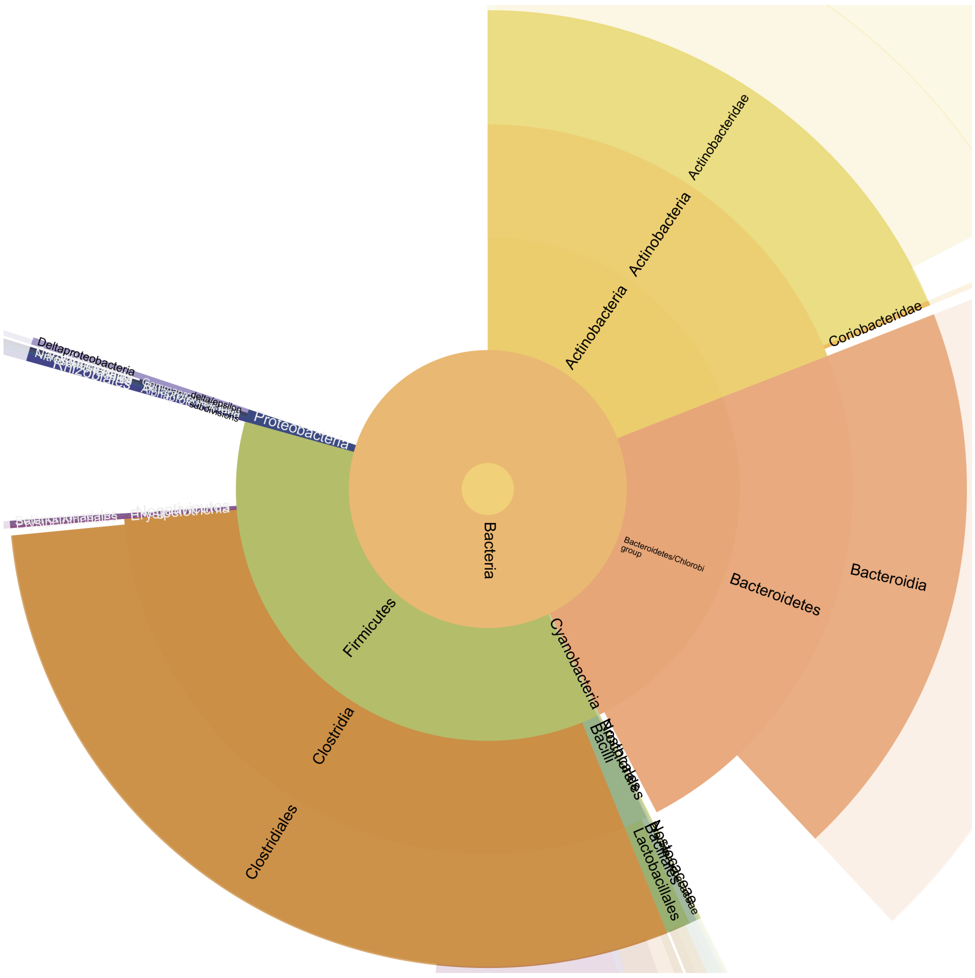


**Figure 2.8** The newly added sunburst visualization available in Unipept when running a multi-peptide analysis on sample 7 as determined by Verberkmoes et al. (2009), zoomed in on Bacteria with default colors.

By default, the colors of the slices are algorithmically grouped to indicate taxonomic clusters. This is done by first assigning a random color from a list of hand picked colors to all of the leaf nodes. The colors of the parent nodes are then recursively calculated by taking the average of the colors of the first two children in the hue-saturation-lightness (HSL) color space. If a node only has a single child, a slightly darker tint of the child color is taken. Although this method of assigning colors is visually pleasing, it is hard to compare sunbursts of several analyses because the colors of the slices depend on the composition of the result set. For this reason, a second coloring option was added to Unipept, which uses a hash function to calculate the color based on the taxonomic data. A hash function is a mathematical function that can be used to convert arbitrary data, in this case the name and rank of taxonomic node, to a strict output format, in this case a color value. By using this approach, a taxonomic node is always colored the same way, regardless of the sample. An example of the comparison of two analyses using the hash-based coloring is shown in Figure 2.9 and Figure 2.10.

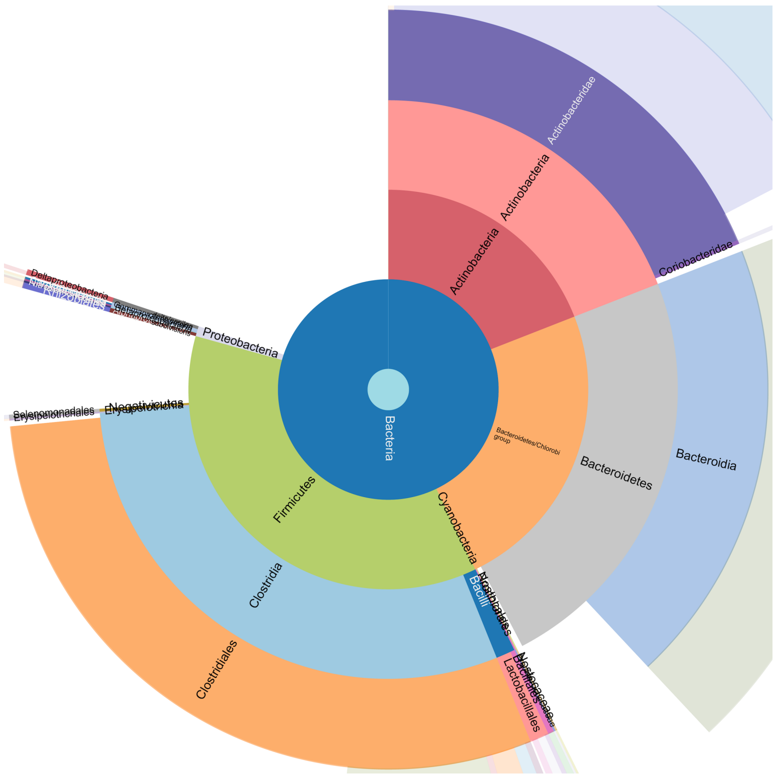


**Figure 2.9** Sunburst diagram visualizing the bacterial diversity in human gut metaproteome sample 7 as determined by Verberkmoes et al. (2009). The diagrams use a fixed color scheme allowing easy comparison of the composition of different samples.

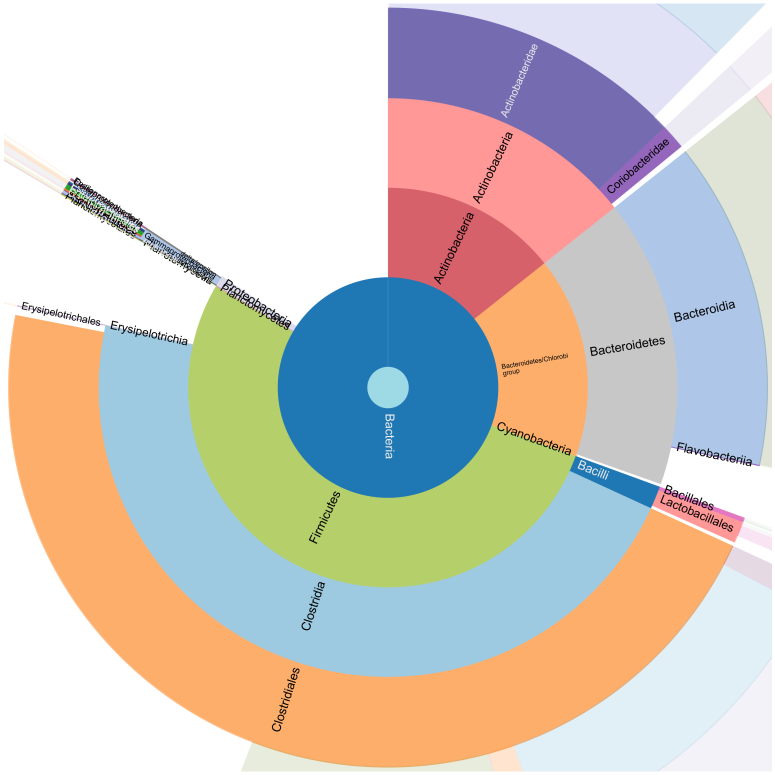


**Figure 2.10** Sunburst diagram visualizing the bacterial diversity in human gut metaproteome sample 8 as determined by Verberkmoes et al. (2009). The diagrams use a fixed color scheme allowing easy comparison of the composition of different samples.

## Treeview

A third way of visualizing the results of a multi-peptide analysis in Unipept is by using an advanced treeview (Figure 2.11). Although the default treeview is great for the visualization of hierarchical data, it falls short for showing the weights (i.e., the number of peptides) associated with the nodes and branches. To overcome this problem, the concept of a Sankey diagram (Riehmann, Hanfler, and Froehlich, 2005) was applied to the treeview. The size of a node now corresponds to the number of peptides associated with that node or any of its children and the width of each branch corresponds to the size of the destination node. As a result, the diameter of each node equals the sum of the width of all outgoing links, supplemented with a proportional share for the number of peptides associated with the node itself. Each of the superkingdoms is assigned a color in which the corresponding nodes and branches are drawn.

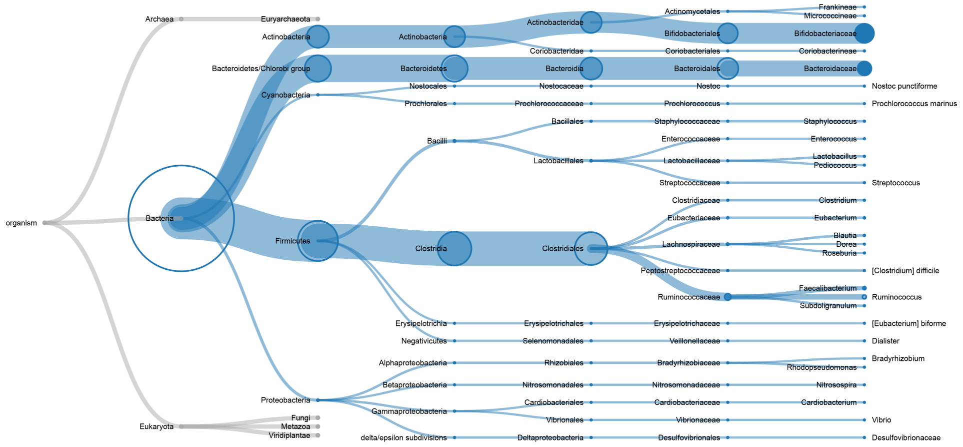


**Figure 2.11** The newly added advanced treeview visualization available in Unipept when running a multi-peptide analysis on sample 7 as determined by Verberkmoes et al. (2009), rescaled on Bacteria.

When the graph initially loads, the root and 2 levels of children are shown. Clicking on a node allows users to expand and collapse the children of that node. In addition, scrolling the mouse invokes a zooming behavior, and dragging lets the graph be repositioned.

The root initially corresponds to 100% of the peptides and is drawn at full size. By right clicking a node of interest, the graph is rescaled to focus on that node. The clicked node is now drawn at full size and all of its children are scaled accordingly. The nodes and branches that are not part of the active subtree are dimmed by drawing them with a thin gray line. Figure 2.11 shows an example of the treeview, focused on Bacteria.

As with the sunburst and treemap visualization, a tooltip with more information is shown when the user hovers over a node. Finally, a full-screen mode and the option to export the visualization as a PNG or SVG image were also added.

### API

The Unipept API and command line interface are discussed in more detail in Chapter 3.

The web-based peptide analysis tools in Unipept are a great fit for exploring the biodiversity of a single sample. However, using the website can be cumbersome when multiple datasets need to be analyzed. To address these issues, all of the peptide analysis functionality (except the visualizations) are also available as a web service. Using this Unipept API, a command line interface to Unipept was implemented in Ruby that allows batch analysis of samples and opens up new ways to include the Unipept functionality in a processing pipeline.

### Peptidome-based analysis

The tryptic peptidome is the complete set of (tryptic) peptides encoded in the genome of an organism. Unipept now provides fast and flexible analysis tools for identifying the unique peptidome of a given taxon and for clustering whole-genomes based on their peptidome content.

The unique peptide finder is discussed in more detail in Section 4.1.

A first tool, the unique peptide finder, computes the unique peptidome for a selected set of RefSeq whole-genome sequences. This unique peptidome consists of all tryptic peptides that are contained in all of the selected genomes, but in none of the UniProt entries belonging to taxa outside those of the selected genomes, making these peptides taxon-specific. Unique peptide sets can be downloaded and used in targeted proteomics experiments.

Peptidome-based clustering is discussed in more detail in Section 4.2.

Peptidome-based clustering computes the UPGMA clustering of a selected set of RefSeq whole-genome sequences based on their peptidome content. Pairwise similarities are computed as the fraction of the size of the intersection over the size of the union of both peptidomes. The results are visualized by a similarity matrix and a phylogenetic tree and can be exported in the Newick and CSV format.

### Website changes

Since usability is a key feature of Unipept, the entire website was redesigned based on several usability studies. A lot of attention was spent on providing an optimal user experience and interaction design. The open source Bootstrap project was used for basic user interface components such as buttons, tabs, and popups and all graphics and visualizations were optimized to take advantage of high-DPI displays.

Another addition to Unipept is an improved export functionality. In addition to the PNG format, Unipept is now also able to export all vector-based visualizations in the SVG file format. Furthermore, CSV exports and copy to clipboard-buttons were added where appropriate for textual results.

The Unipept Multi-peptide Analysis now also directly interfaces with PRIDE (Vizcaíno et al., 2013) to allow a one-click analysis of data deposited in the PRIDE archive. Users can simply enter the ID of any PRIDE experiment and click the “Load Dataset” button. Unipept subsequently fetches the data and preloads the search form for further analysis.

### Open source

The source code and documentation of the current and all previous Unipept releases have been made available on GitHub at http://GitHub.com/unipept/unipept. All code is licensed under the permissive MIT license. By open sourcing Unipept, we provide optimal transparency on how our algorithms work and invite other researchers to contribute to Unipept. Additionally, the source code of individual components (e.g., the visualizations) may be extracted for use in other projects. Another advantage of making all code available, is that research groups may set up their own Unipept server. Setting up a local Unipept server provides a solution for researchers who are not entitled to send out proprietary data to public, third party services. Running a local server also limits the dependency on external resources, which was a stumbling block for several users to include Unipept in their workflow.

### 2.2.3 Statistics

The initial Unipept release was based on UniProt 2012_07 containing 17 million protein entries. Unipept 2.3 is based on UniProt 2014_05 containing 56.5 million protein entries. In 22 months, the amount of data in UniProt more than tripled and Unipept now contains information on over a billion peptides. The number of distinct tryptic peptides went up from 250 million to 402 million. Unipept now also parses additional protein metadata such as EC numbers (Bairoch, 2000) and Gene Ontology (Ashburner et al., 2000) annotations and displays this information on the single peptide analysis results page. The increase in data caused the size of the Unipept database to grow from 81 GB (39 GB data, 42 GB index) to 188 GB (91 GB data, 97 GB index). The exponential growth in data poses some challenges for the future regarding data processing and storage. An incremental update strategy, next to the use of high performance key-value stores such as Berkeley DB (Olson, Bostic, and Seltzer, 1999) or in memory databases like VoltDB (Stonebraker and Weisberg, 2013), might offer some solutions to reduce the time needed to parse UniProt.

The next data processing pipeline used a Berkeley DB-based solution to speed up parsing.

The analysis on the Unipept website is not negatively affected by the growing data, on the contrary, the results become increasingly more accurate.

Unipept uses the NCBI Taxonomy as its reference taxonomy database. The taxonomic information in Unipept is only updated when a new UniProt version is parsed. These updates sometimes introduce erroneous taxa such as the species "*metagenomes*" (NCBI taxon ID 256318). By adjusting the invalidation algorithms, the impact of such taxa can be reduced, but this requires manual inspection with each update.

The number of Unipept users is growing each month, with over a thousand users in June 2014. In total, 4000 multi-peptide and 3000 single peptide analyses were performed, accounting for 81 million processed peptides.

## 2.3 Recent additions

After the publication in Proteomics (Mesuere et al., 2015), we didn't stop adding new features to the diversity analysis in Unipept. On a data level, the biggest update was the implementation of a new data processing pipeline as already mentioned in the previous section. The new pipeline can process UniProt in under 7 hours as opposed to almost 3 months for the previous one. This enables us to incorporate the monthly UniProt updates in a timely manner. The UniProt version that is used by Unipept is now also displayed in the footer of each page.

The Tryptic Peptide Analysis was expanded with a few minor changes. The treeview that was introduced in the Metaproteome Analysis was open sourced on GitHub (https://github.com/unipept/unipept-visualizations) as a stand-alone visualization. This allowed us to replace the existing tree visualization in the Tryptic Peptide Analysis with our own new treeview. Additionally, the table displaying all UniProt matches was supplemented with the name of the proteins, an *open in UniProt* button and a *copy to clipboard* button.

The visualizations of the Metaproteome Analysis underwent bigger changes. The treemap was entirely rewritten using D3, allowing more customization such as the addition of a breadcrumb bar. Major parts of the sunburst graph were also rewritten and an enhanced breadcrumbs bar was added to the visualization. The fairly new treeview gained a shift-click option that expands all children of the clicked node. Additionally, a few bugs were fixed improving the layout of the graph and eliminating overlapping nodes. These improvements allowed for a new full screen mode that enables fast switching between all three visualizations.

# Chapter 3

# The Unipept API and command line tools

The Unipept website is an excellent tool for biodiversity analysis of metaproteomics samples. One disadvantage for large-scale data processing is that all analysis must be run manually on the website. For high-throughput studies, this manual approach is not viable. With this in mind, we developed an API and a set of command line tools to expose the Unipept analysis features for use in automated pipelines and other applications.

In this chapter, we first describe the Unipept API by means of the application note published in Bioinformatics (Mesuere et al., 2016b) and then give an overview of the command line tools with two detailed case studies.

## 3.1 Unipept web services for metaproteomics analysis

This section contains a verbatim copy of the application note by Mesuere et al. (2016b) as published in Bioinformatics.

**Abstract** — Unipept is an open source web application that is designed for metaproteomics analysis with a focus on interactive data-visualization. It is underpinned by a fast index built from UniProtKB and the NCBI taxonomy that enables quick retrieval of all UniProt entries in which a given tryptic peptide occurs. Unipept version 2.4 introduced web services that provide programmatic access to the metaproteomics analysis features. This enables integration of Unipept functionality in custom applications and data processing pipelines. The web services are freely available at http://api.unipept.ugent.be and are open sourced under the MIT license.

### 3.1.1 Introduction

Unipept is a web application for biodiversity analysis of complex metaproteomics samples (Mesuere et al., 2012). The application is powered by a fast index built from UniProtKB (The Uniprot Consortium, 2015) and a cleaned up version of the NCBI taxonomy (Federhen, 2012). This index enables quick retrieval of all UniProt entries in which a given tryptic peptide occurs. Using the taxonomic annotations of UniProt entries, Unipept also returns the complete set of organisms in which a given peptide occurs. This set of organisms is then processed using a Lowest Common Ancestor (LCA) algorithm to determine the taxonomic specificity of the peptide. All results are presented in clear overview tables and in an interactive treeview.

Fast computation of LCAs for given lists of peptides also enables interactive biodiversity analysis of metaproteomics data sets. The biodiversity in complex samples can then be inspected using multiple interactive visualizations such as a treeview (Figure 3.1), a sunburst view (Figure 3.2) and a treemap (Figure 3.3). All visualizations on the Unipept website can be saved as publication-grade graphics, and all analysis results can be exported as Microsoft Excel-compatible CSV files.

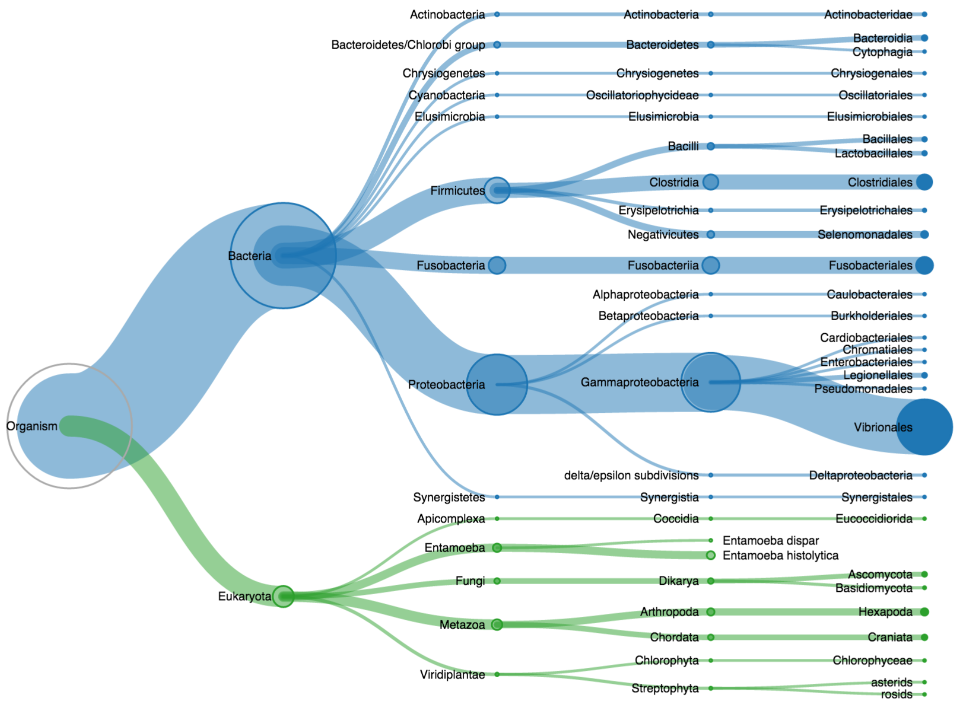


**Figure 3.1** Interactive treeview visualization that shows the tree of the matched taxa resulting from a Tryptic Peptide Analysis on the tryptic peptide AEAHIK.

To guarantee optimal performance and correctness, the Unipept project pursues excellence regarding best practices for modern web application development. One example of this is automatic correctness testing by over 1000 tests after each code change. The entire application including the web services is open source and licensed under the terms of the MIT license. The source code can be found at http://GitHub.com/unipept/unipept.

In this article, we present the latest addition to the Unipept toolbox: a set of web services that expose the Unipept analysis functions for use in other applications and data processing pipelines.

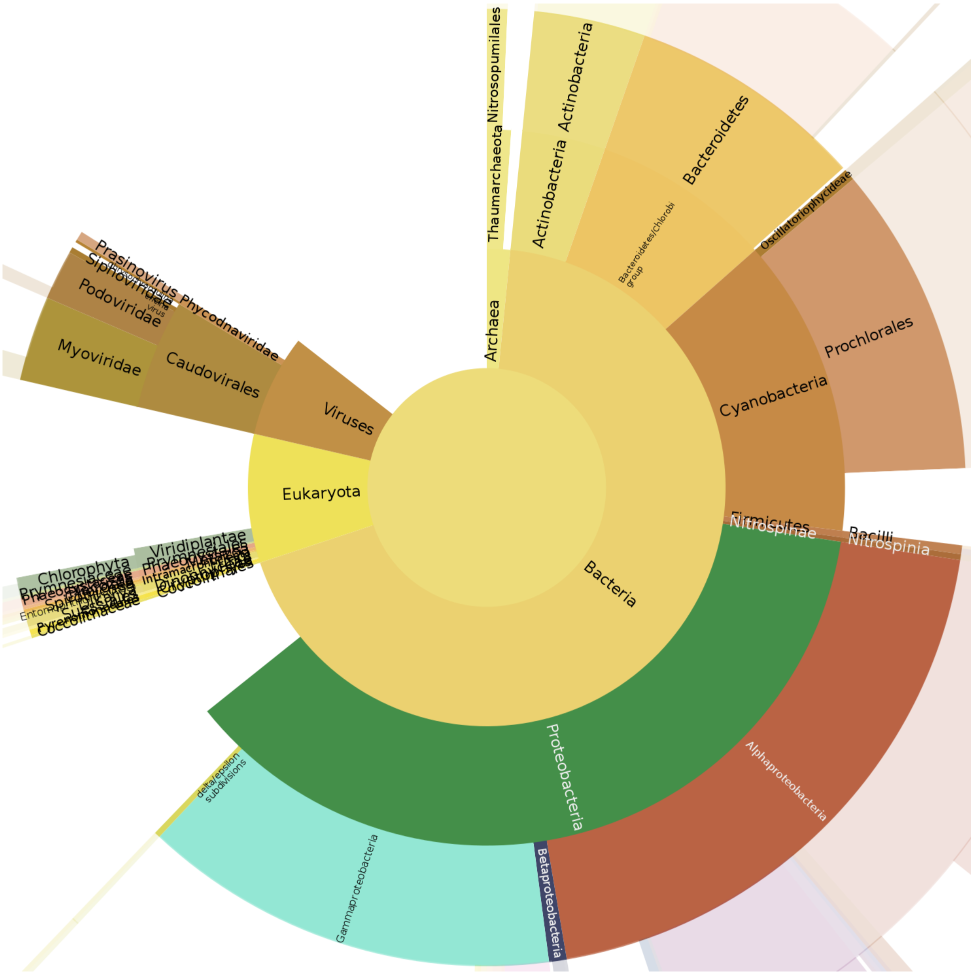


**Figure 3.2** Example of the interactive sunburst visualization in Unipept. The graph shows the result of the marine sample dataset available on the Unipept website.

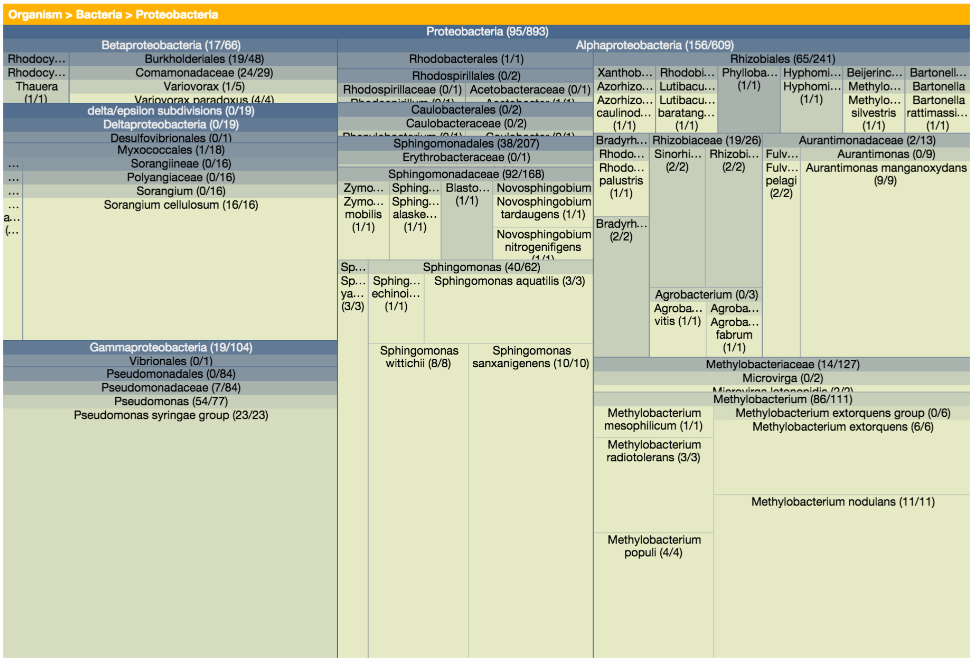


**Figure 3.3** Example of the interactive treemap in Unipept. The graph shows the result of the *Arabidopsis thaliana* phyllosphere sample dataset available on the Unipept website.

## 3.1.2 Methods

Unipept version 2.4 introduced the Unipept web services. These web services allow access to all Unipept peptide analysis features through a RESTful API. This means that all communication with the web services can be done using simple stateless HTTP requests, to which the server answers in JSON. JSON is an open standard for transmitting data that is both human readable and has wide support in developer tools and programming languages.

In the next sections, we discuss the available API functions by drawing parallels between usage of the Unipept website and the Unipept API. Figure 3.4 displays a schematic overview of the included functions, along with the expected input and output. The full documentation can be found at http://api.unipept.ugent.be. Next to the documentation, the website also offers an interactive API explorer (Figure 3.5) where API requests can be composed and tested with just a few clicks.

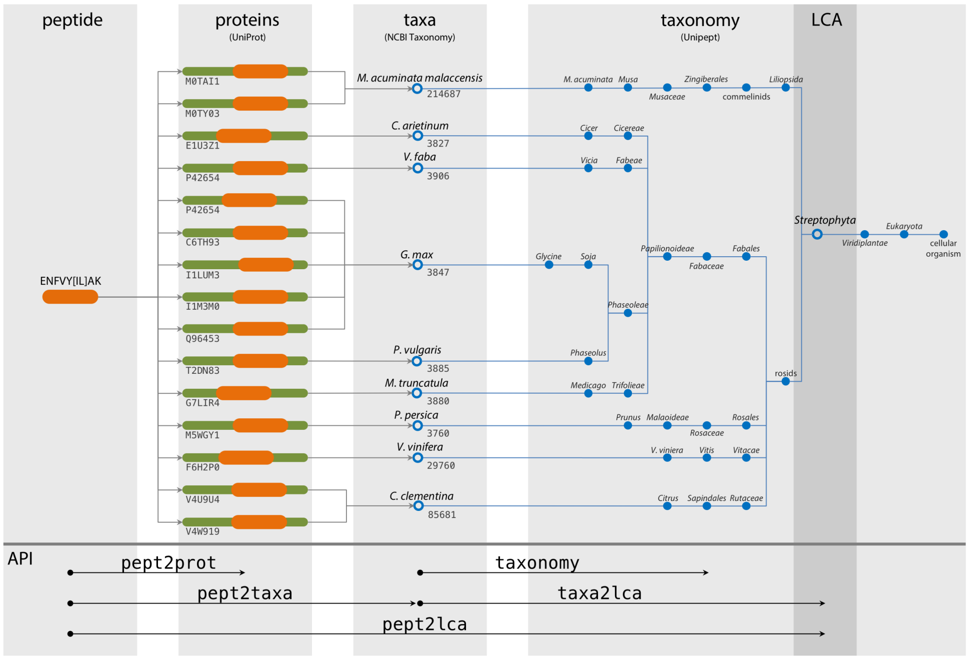


**Figure 3.4** General outline of the Unipept workflow for taxonomic identification of tryptic peptides. For a given tryptic peptide, all UniProt entries having an exact match of the peptide in the protein sequence are found. Unipept then computes the lowest common ancestor (LCA) of the taxonomic annotations extracted from the matched UniProt entries, based on a cleaned up version of the NCBI Taxonomy. All intermediate results are shown for the sample tryptic peptide ENFVY[IL]AK (isoleucine and leucine equated), leading to an LCA in the phylum Streptophyta. Arrows at the bottom show which processing steps are available as functions in the Unipept API.

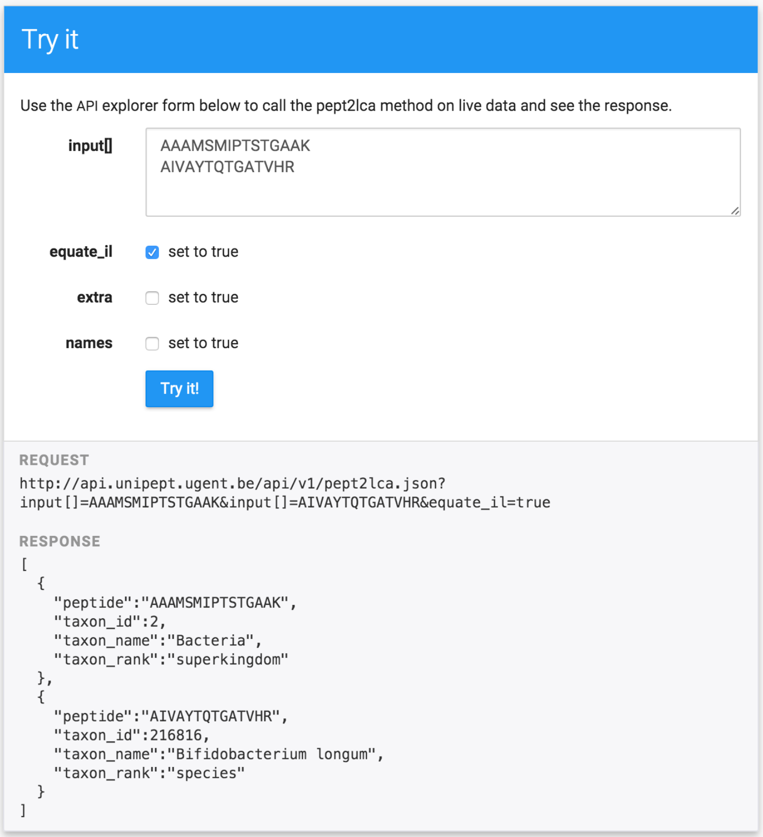
## pept2prot

The fundamental component in the Tryptic Peptide Analysis feature of Unipept is fast retrieval of all UniProt entries in which a given tryptic peptide occurs. All subsequent calculations are based on this result, and therefore the database indexes are heavily optimized to return the result as fast as possible. When doing a Tryptic Peptide analysis in the web interface, the set of all matching UniProt entries is listed on the Protein Matches tab.

Its web service counterpart, `pept2prot`, takes a single tryptic peptide as input and returns the list of all UniProt entries containing the given tryptic peptide. By default, for each entry, the UniProt accession number, protein name and associated NCBI taxon ID are returned. Optionally, users can also request additional information fields such as the name of the organism associated with the UniProt entry, a list of cross-referenced EC numbers (Bairoch, 2000) and a list of cross-referenced GO terms (The Gene Ontology Consortium, 2014). Users can also choose to equate the isobaric amino acids isoleucine (I) and leucine (L) when matching peptides to proteins, a typical option for mass spectrometry-related queries. Batch retrieval of multiple peptides at once is also supported.

#### pept2taxa

After matching the UniProt entries, Unipept uses the cross-referenced NCBI taxon IDs to compile a set of organisms in which the queried peptide occurs. These organisms are then mapped to their taxonomic lineages using a cleaned up version of the NCBI taxonomy database. Using the web interface, the list of organisms along with their lineage can be found in the Lineage Table tab and an interactive visualization is available in the Lineage Tree tab (Figure 3.1).



Use the API explorer form below to call the pept2lca method on live data and see the response.

| | |
|---|---|
| input[] | AAAMSMIPTSTGAAK<br>AIVAYTQTGATVHR |
| equate_il | [✓] set to true |
| extra | [ ] set to true |
| names | [ ] set to true |

REQUEST

```
http://api.unipept.ugent.be/api/v1/pept2lca.json?
input[]=AAAMSMIPTSTGAAK&input[]=AIVAYTQTGATVHR&equate_il=true
```

RESPONSE

```
[
  {
    "peptide":"AAAMSMIPTSTGAAK",
    "taxon_id":2,
    "taxon_name":"Bacteria",
    "taxon_rank":"superkingdom"
  },
  {
    "peptide":"AIVAYTQTGATVHR",
    "taxon_id":216816,
    "taxon_name":"Bifidobacterium longum",
    "taxon_rank":"species"
  }
]
```

**Figure 3.5** Screenshot of the API explorer, available on the documentation page of each of the Unipept API functions. By using the form, all of the API features can be easily tested within a web browser. After clicking the "Try it!" button, the resulting query string and response are shown. The figure shows the output for the `pept2lca` method used on the tryptic peptides AAAMSMIPTSTGAAK and AIVAYTQTGATVHR with the option to equate isoleucine and leucine when matching peptides to proteins. The LCA for AAAMSMIPTSTGAAK is the superkingdom Bacteria, the LCA for AIVAYTQTGATVHR is the species *Bifidobacterium longum*.

Similarly, the API function `pept2taxa` takes a tryptic peptide as input and returns the set of organisms associated with the UniProt entries containing the given tryptic peptide. By default, the taxon ID, name, and rank are returned for each of the matched organisms. Optionally, the full lineage of each organism can be requested as a sequence of taxon IDs and/or taxon names. Batch requests and equating isoleucine and leucine are also supported.

### pept2lca

The matched organisms from the previous section are then used to calculate the taxonomic lowest common ancestor (LCA). Simply put, the LCA is the most specific taxonomic rank that all matched organisms have in common. However, the algorithm used by Unipept has several advancements to better cope with taxonomic noise and misclassifications (Mesuere et al., 2012). One of these improvements is the invalidation of taxonomic nodes that provide little informational values, such as those containing words like “uncultured”, “unspecified” or “undetermined” in their name. Invalidated taxa are ignored during LCA calculation and mapped to their first valid ancestor. These invalidated taxa would otherwise result in a drastic loss of information when used for LCA calculation. Another example is mapping strain-specific taxon IDs to their first valid parent taxon to counter the, now abandoned, practice of creating strain-level taxon IDs (Federhen et al., 2014).

Newer versions of the invalidation script also contain exceptions for viruses.

Correspondingly, the `pept2lca` function returns the LCA (taxon ID, name, and rank) for a given tryptic peptide. Optionally, the full lineage (IDs and/or names) can be requested and both equating isoleucine and leucine and batch requests are supported. The LCAs for all tryptic peptides are precalculated and stored in the database. Therefore, the peptide matching steps can be skipped for the pept2lca function, resulting in improved performance.

### taxa2lca

The Unipept LCA algorithm can also be used outside a proteomics context by using the `taxa2lca` function. This API function takes a list of NCBI taxon IDs and calculates their LCA by using the advanced algorithm as applied by Unipept. The result is returned by listing the taxon ID, name, and rank of the LCA. Additional lineage information is also available upon request. Note that the `pept2lca` function can be mimicked by chaining the `pept2taxa` and `taxa2lca` functions. This is however not recommended, as `pept2lca` makes use of precomputed data and is therefore several orders of magnitude faster.

**taxonomy**

The `taxonomy` function provides access to the cleaned up version of the NCBI taxonomy as used by Unipept. This function can be used, for example, to compute more detailed statistics about taxon hits or implement alternative aggregation strategies next to the LCA computation as used by Unipept. The function takes one or more taxon IDs as input and returns the name and rank for each of the given IDs. Optionally, the full lineage can also be returned.

### 3.1.3 Results

The Unipept project consists of two main parts: a collection of scripts to construct the database and the web application. The first part of the database construction, the code to parse UniProt, was recently updated to use Berkeley DB (Olson, Bostic, and Seltzer, 1999), a high performance key value store, to store intermediate results. This resulted in an enormous boost in parsing speed: where the old parser took over 30 days to parse UniProt, the new approach using Berkeley DB does the job in under 10 hours. The second part of the database construction is the precalculation of the LCAs of all the peptides in the database. The old Ruby code was rewritten in Java and computation time was reduced from over four weeks to just 15 minutes with the help of some new Java 8 features. The combination of these advancements allows us to consistently offer analysis results based on the latest UniProt release. The second part of Unipept is a Ruby on Rails web application that uses JavaScript for all client side interactions. All data visualizations (Mesuere et al., 2015) are made in-house with the D3.js JavaScript library (Bostock, Ogievetsky, and Heer, 2011).

The GalaxyP project already takes advantage of the new Unipept web services to integrate Unipept functionality into the Galaxy Framework (Jagtap et al., 2015). Exact matching of peptides to UniProt entries is also implemented by the Peptide Match application (Chen et al., 2013) of the Protein Information Resource (PIR). Where Unipept is restricted for use with tryptic peptides, Peptide Match has no such limitation. However, the advantage of accepting all peptides comes at the cost of reduced performance. For a test set of 500 tryptic peptides, the Unipept `pept2prot` function returned all matching UniProt entries in 1.5 seconds whereas Peptide Match took over 33 minutes. Since (meta)proteomics experiments almost exclusively use trypsin to digest proteins, resulting in a list of tryptic peptides, this is a reasonable compromise (Olsen, Ong, and Mann, 2004). All functions of the Unipept API are tweaked for optimal performance and usable for high throughput data analysis. The `pept2lca` function (no counterpart in PIR), can process over 10 000 peptides per second. For this reason, the information fields that are returned by default are limited to the subset of available fields that can be returned without performance penalty.

## 3.2 The Unipept command line tools

The Unipept API provides a good starting point for integrating Unipept functionality in other applications and pipelines. It is however not a ready-made solution that can be used without additional programming. To counter this, we developed a set of user-friendly command line tools that are essentially wrappers around the API. In this section, we first give an overview of the Unipept command line tools and their advantages over the API and then demonstrate their usage in two detailed case studies.

### 3.2.1 The Unipept gem

The Unipept command line tools provide a command line interface to the Unipept web services along with a few utility commands for handling proteins using the command line. All tools support fasta and plain text input, multiple output formats (csv, xml, and json) and parallel web requests for improved performance. Just as with the Unipept web application, we followed the coding best practices and the entire code base is covered with unit and integration tests. All code is open source under the MIT License and available on GitHub in a separate `unipept-cli` repository at https://GitHub.com/unipept/unipept-cli.

#### Installation

The Unipept command line tools are written in Ruby, so to use them, Ruby needs to be installed on your system. We recommend using Ruby 2.2, but all versions since Ruby 1.9.3, as well as JRuby are supported. We made the command line tools available as a Ruby gem. A gem is a packaged version of the code that can be used in combination with the RubyGems package manager. This means it can easily be installed with a single command:

```
$ gem install unipept
Fetching: unipept-1.1.0.gem (100%)
Successfully installed unipept-1.1.0
Parsing documentation for unipept-1.1.0
Installing ri documentation for unipept-1.1.0
Done installing documentation for unipept after 0 seconds
1 gem installed
```

After successful installation, the unipept command should be available. To check if the gem was installed correctly, run `unipept --version`. This should print the version number:

```
$ unipept --version
1.1.0
```

Updating to the newest version of the command line tools is equally simple using the `gem update unipept` command. Each of the commands also has a built-in help function that can be displayed using the `--help` argument.

```
$ unipept --help
NAME
unipept - Command line interface to Unipept web services.

USAGE
unipept subcommand [options]

DESCRIPTION
The unipept subcommands are command line wrappers around the Unipept web
services.

Subcommands that start with pept expect a list of tryptic peptides as
input. Subcommands that start with tax expect a list of NCBI Taxonomy
Identifiers as input. Input is passed
- as separate command line arguments
- in a text file that is passed as an argument to the -i option
- to standard input
```

```
The command will give priority to the first way the input is passed, in
the order as listed above. Text files and standard input should have one
tryptic peptide or one NCBI Taxonomy Identifier per line.

COMMANDS
config        Set configuration options.
help          show help
pept2lca      Fetch taxonomic lowest common ancestor of UniProt entries
              that match tryptic peptides.
pept2prot     Fetch UniProt entries that match tryptic peptides.
pept2taxa     Fetch taxa of UniProt entries that match tryptic peptides.
taxa2lca      Compute taxonomic lowest common ancestor for given list of
              taxa.
taxonomy      Fetch taxonomic information from Unipept Taxonomy.

OPTIONS
-f --format=<value>          define the output format (available: json,
                             csv, xml) (default: csv)
-h --help                    show help for this command
   --host=<value>            specify the server running the Unipept web
                             service
-i --input=<value>           read input from file
-o --output=<value>          write output to file
-q --quiet                   disable service messages
-v --version                 displays the version
```

### Commands

The Unipept command line tools consist of four main commands: `uniprot`, `prot2pept`, `peptfilter`, and `unipept`.

The `uniprot` command is a utility to easily fetch protein information from UniProt. It takes one or more UniProt accession numbers and returns the corresponding UniProt entry for each of the accession numbers as output. This information is fetched by using the UniProt web services.

```
$ uniprot C6JD41 Q06JG4
MTLVPLGDRVVLKQVEAEETTKSGIVLPGQAQEKPQQAEVVAVGPGGVVDGKEVKMEVAVGDKVIYSKYSGT
  EVKMDGTEYIIVKQNDILAIVK
MFTNSIKNLIIYLMPLMVTLMLLSVSFVDAGKKPSGPNPGGNN
```

`prot2pept` is a utility to perform an *in silico* trypsin digest on protein sequences. The command takes one or more protein sequences as input and returns the digested peptides as output. This command runs entirely locally and doesn't connect to any server.

```
$ echo "LGAARPLGAGLAKVIGAGIGIGK" | prot2pept
LGAARPLGAGLAK
VIGAGIGIGK
```

The `peptfilter` command also runs entirely client-side and can be used to filter a list of peptides that satisfy a given set of criteria. By default, peptides with length between 5 and 50 are retained, but other criteria can be specified. The length filter can be changed by using the `--minlen` and `--maxlen` parameters. Peptides can also be filtered based on whether or not they lack or contain certain amino acids. This can be done with the `--lacks` and `--contains` parameters.

```
$ cat input.txt
AAR
AALTER
$ cat input.txt | peptfilter
AALTER
```

The `unipept` command has several subcommands: `pept2lca`, `pept2taxa`, `pept2prot`, `taxa2lca`, and `taxonomy`. Each of them corresponds to the equally named API call and has several options that are equivalent with the corresponding API parameters. A comprehensive description of these calls and parameters can be found in Section 3.1.2 of this chapter, examples of their usage can be found in the case studies in Section 3.2.2 and Section 3.2.3.

### Input and output formats

One of the benefits of using the command line tools is support for multiple input and output formats. All commands accept input from command line arguments, a file, or *standard input*. Output can be written to *standard output* or to a file. Where the API always uses json as output format, the command line tools offer support for json, xml, and csv. Additionally, the `--select` option allows you to control which fields are returned. A list of fields can be specified by a comma-separated list, or by using multiple `--select` options. The ∗-symbol can be used as a wildcard for field names. For example, `--select peptide,taxon*` will return the `peptide` field and all fields starting with `taxon`.

The commands also support input (from any source) in fasta(-like) format. This format consists of a fasta header (a line starting with a >-symbol), followed by one or more lines containing, for example, one peptide each. When this format is detected, the output will automatically include an extra information field in the output containing the corresponding fasta header.

```
$ cat input.txt
> header 1
AALTER
MDGTEYIIVK
> header 2
AALTER
$ unipept pept2lca --input input.txt
fasta_header,peptide,taxon_id,taxon_name,taxon_rank
> header 1,AALTER,1,root,no rank
> header 1,MDGTEYIIVK,1263,Ruminococcus,genus
> header 2,AALTER,1,root,no rank
```

### Request optimizations

Internally, the command line tools query the Unipept API for each of the `unipept` subcommands. Making one http-request at a time for each of the input values would cause a significant overhead and poor performance. This problem was tackled in two ways. A first solution is batching multiple input values per request. The batch size was determined experimentally and lies between 10 and 1000 values per request, depending on the subcommand and the amount of extra information that is requested.

A second solution is doing multiple parallel requests at a time to make optimal use of the multiple CPU cores of the client and server. This fix improves performance significantly, but has an unexpected side effect: because multiple requests are sent at a time, we don't know in what order they will be completed. To make sure that the order of the input and output values are the same, we had to include a non-trivial reordering algorithm to restore the order without consuming lots of memory.

## 3.2.2 Case study: analysis of a tryptic peptide

For a first case study, let's say that we have determined the mass spectrum of a tryptic peptide, that was identified as the peptide ENFVYIAK using database searches (*Mascot* (Perkins et al., 1999), *Sequest* (Eng, McCormack, and Yates, 1994), *X!Tandem* (Craig and Beavis, 2003)) or de novo identification (*PEAKS* (Ma et al., 2003)). As an example, we show how this tryptic peptide can be taxonomically assigned to the phylum Streptophyta. As a starter, we can use the `unipept pept2prot` command to fetch all UniProt proteins indexed by Unipept that contain the peptide.

#### Matching proteins

The following interactive session shows that UniProt contains 19 proteins that contain the tryptic peptide ENFVYIAK. Note that the first command passes the tryptic peptide as an argument to the `unipept pept2prot` command. In case no tryptic peptide is passed as an argument, the command reads a tryptic peptide from standard input as illustrated by the second command. Throughout this case study we will preferentially pass tryptic peptides as an argument to the `unipept pept2prot` command, but the command works the same way irrespective of how the tryptic peptide is fed to the command.

```
$ unipept pept2prot ENFVYIAK
peptide,uniprot_id,protein_name,taxon_id
ENFVYIAK,C6TH93,Casparian strip membrane protein 4,3847
ENFVYIAK,P42654,14-3-3-like protein B,3906
ENFVYIAK,Q96453,14-3-3-like protein D,3847
...
$ echo "ENFVYIAK" | unipept pept2prot
peptide,uniprot_id,protein_name,taxon_id
ENFVYIAK,C6TH93,Casparian strip membrane protein 4,3847
ENFVYIAK,P42654,14-3-3-like protein B,3906
ENFVYIAK,Q96453,14-3-3-like protein D,3847
...
```

By default, the output is generated in csv-format (*comma-separated values*). Apart from the query peptide (`peptide`), the output contains two GUIDs (*globally unique identifiers*): i) the UniProt Accession Number (`uniprot_id`) that refers to the protein record in the UniProt database that contains the tryptic peptide and ii) the NCBI Taxonomy Identifier (`taxon_id`) assigned to the UniProt protein record that refers to a record in the NCBI Taxonomy Database (Acland et al., 2014; Benson et al., 2013). The latter describes a taxon in the hierarchical classification of cellular organisms, being the taxon from which the protein was extracted. The output also contains the name of each protein (`protein_name`).

### Leucine & isoleucine

In peptide sequencing experiments involving a single step tandem mass acquisition, leucine (`L`) and isoleucine (`I`) are indistinguishable because both are characterized by a 113 Da mass difference from the other peptide fragments in the MS-MS spectrum. In general, there are $2^n$ `I=L` variants for each tryptic peptide that contains *n* residues that are either leucine or isoleucine. Therefore, all subcommands of the unipept command that are based on matching given peptides against UniProt proteins support the `-e/--equate` option. Exact matching makes no distinction between `I` and `L` when this option is activated.

```
$ unipept pept2prot -e ENFVYIAK
peptide,uniprot_id,protein_name,taxon_id
ENFVYIAK,C6TH93,Casparian strip membrane protein 4,3847
ENFVYIAK,P42654,14-3-3-like protein B,3906
ENFVYIAK,Q96453,14-3-3-like protein D,3847
ENFVYIAK,G7LIR4,Uncharacterized protein,3880
...
```

Note that the Unipept database has two separate index structures to match tryptic peptides against UniProt protein records: one that is used to exactly match tryptic peptides against UniProt protein records and one that is used to exactly match all `I=L` variants of a given tryptic peptide. As a result, matching all `I=L` variants of the tryptic peptide ENFVYIAK can be done in a single step, without any performance loss.

### Metadata

Apart from a fast index that maps tryptic peptides onto the UniProt entries of proteins that contain the peptide, the Unipept database contains minimal information about the proteins that was extracted from the UniProt entries. This includes information about the taxon from which the protein was sequenced (`taxon_id` and `taxon_name`) and a description of the cellular functions the protein is involved in (`ec_references` and `go_references`). Taxonomic information is described using a GUID that refers to a record in the NCBI Taxonomy Database (Acland et al., 2014; Benson et al., 2013). Functional information is described using GUIDs that refer to records from the Enzyme Commission classification (EC; Bairoch (2000)) and the Gene Ontology (GO; Ashburner et al. (2000)). The generated output contains this additional information if the `-a/--all` option of the `unipept` command is used. The following example is representative in the sense that the taxonomic information about proteins is generally more accurate and complete than the information about known functions of the proteins.

```
$ unipept pept2prot -e -a ENFVYIAK
peptide,uniprot_id,protein_name,taxon_id,taxon_name,ec_references,go_ref
  erences,refseq_ids,refseq_protein_ids,insdc_ids,insdc_protein_ids
ENFVYIAK,C6TH93,Casparian strip membrane protein 4,3847,Glycine max,,GO:
  0016021 GO:0005886 GO:0071555,NM_001255156.1,NP_001242085.1,BT097011,A
  CU21195.1
ENFVYIAK,P42654,14-3-3-like protein B,3906,Vicia faba,,,,,Z48505,CAA8841
  6.1
ENFVYIAK,Q96453,14-3-3-like protein D,3847,Glycine max,,,NM_001250136.1,
  NP_001237065.1,U70536,AAB09583.1
ENFVYIAK,G7LIR4,Uncharacterized protein,3880,Medicago truncatula,,,XM_00
  3629715.1,XP_003629763.1,CM001224 BT141273,AET04239.2 AFK41067.1
...
```

Because Unipept uses a separate peptide index in which `I` and `L` are equated, Unipept cannot directly resolve what specific `I=L` variant (or variants) of a tryptic peptide are contained in a protein sequence. However, the Unipept command line tools contain the `uniprot` command that calls the UniProt web services. This can be used, for example, to retrieve all protein sequences for a given list of UniProt Accession Numbers. The following example also illustrates the `-s/--select` option of the `unipept` command, that can be used to include only a selected list of information fields in the generated output. Note that we add a series of additional processing steps to the result of the `uniprot` command, that only put the contained `I=L` variants in capitals (the remaining residues are converted into lower case) and truncate the protein sequences after a fixed number of residues.

```
$ unipept pept2prot -e ENFVYIAK -s uniprot_id | tail -n+2 | \
  uniprot | tr 'A-Z' 'a-z' | sed 's/enfvy[il]ak/\U&\E/' | \
  sed -E 's/(.{60}).*/\1.../'
maaskdrENFVYIAKlaeqaeryeemvesmknvanldveltveerkkgvaildfilrlga...
mastkdrENFVYIAKlaeqaeryeemvdsmknvanldveltieernllsvgyknvigarr...
mtaskdrENFVYIAKlaeqaeryeemvesmknvanldveltveernllsvgyknvigarr...
mastkerENFVYIAKlaeqaeryeemveamknvakldveltveernllsvgyknvvgahr...
mdkdrENFVYIAKlaeqaerydemvdamkkvanldveltveernllsvgyknvigarras...
...
```

The `uniprot` command can not only be used to fetch protein sequences from the UniProt database, but also all metadata that is available about the protein in UniProt. This can be done by passing a specific format to the `-f/--format` option of the `uniprot` command: `csv` (default value), `fasta`, `xml`, `text`, `rdf` or `gff`. As an example, the following session fetches the first three proteins from UniProt that contain an `I=L` variant of the tryptic peptide ENFVYIAK. These proteins are returned in FASTA format.

```
$ unipept pept2prot -e ENFVYIAK -s uniprot_id | tail -n+2 | \
  head -3 | uniprot -f fasta
>sp|C6TH93|CASP4_SOYBN Casparian strip membrane protein 4 OS=Glycine max
MAASKDRENFVYIAKLAEQAERYEEMVESMKNVANLDVELTVEERKKGVAILDFILRLGA
ITSALGAAATMATSDETLPFFTQFFQFEASYDSFSTFQFFVIAMAFVGGYLVLSLPFSIV
TIIRPHAAGPRLFLIILDTVFLTLATSSAAAATAIVYLAHNGNQDSNWLAICNQFGDFCQ
EISGAVVASFVAVVLFVLLIVMCAVALRNH
>sp|P42654|1433B_VICFA 14-3-3-like protein B OS=Vicia faba PE=2 SV=1
MASTKDRENFVYIAKLAEQAERYEEMVDSMKNVANLDVELTIEERNLLSVGYKNVIGARR
ASWRILSSIEQKEESKGNDVNAKRIKEYRHKVETELSNICIDVMRVIDEHLIPSAAAGES
TVFYYKMKGDYYRYLAEFKTGNEKKEAGDQSMKAYESATTAAEAELPPTHPIRLGLALNF
SVFYYEILNSPERACHLAKQAFDEAISELDTLNEESYKDSTLIMQLLRDNLTLWTSDIPE
```

```
DGEDSQKANGTAKFGGGDDAE
...
```

### Lowest common ancestor

Based on the taxonomic annotations contained in the UniProt entries that match a given tryptic peptide, the tryptic peptide can be assigned taxonomically. To do so, Unipept makes use of an algorithm that computes the *lowest common ancestor* (LCA) of all taxa in which the peptide was found. The implementation of this algorithm in Unipept is robust against taxonomic misarrangements, misidentifications, and inaccuracies. Unipept computes the LCA based on the *Unipept Taxonomy*, a cleaned up version of the NCBI Taxonomy that heuristically invalidates some "unnatural" taxa from the original database based on a set of regular expressions. Not taking into account this identification noise would otherwise result in drastic loss of information.

Apart from the LCA algorithm implemented by Unipept, it is also possible to come up with alternative aggregation scenarios that are implemented client side based on the NCBI Taxonomy Identifiers that are associated with the matched UniProt protein records. Scenarios that are based on the Unipept Taxonomy can be implemented by using the `unipept pept2taxa` command that outputs all taxa associated with the UniProt proteins that contain a given tryptic peptide.

```
$ unipept pept2taxa -e ENFVYIAK
peptide,taxon_id,taxon_name,taxon_rank
ENFVYIAK,2711,Citrus sinensis,species
ENFVYIAK,3760,Prunus persica,species
ENFVYIAK,3827,Cicer arietinum,species
ENFVYIAK,3847,Glycine max,species
...
```

Using the `-a` option in combination with the `unipept pept2taxa` command includes the complete lineages (resulting after the cleanup done by Unipept) of the taxa in the generated output.

```
$ unipept pept2taxa -e -a ENFVYIAK
peptide,taxon_id,taxon_name,taxon_rank,superkingdom_id,superkingdom_name
ENFVYIAK,2711,Citrus sinensis,species,2759,Eukaryota,33090,Viridiplantae
ENFVYIAK,3760,Prunus persica,species,2759,Eukaryota,33090,Viridiplantae,
ENFVYIAK,3827,Cicer arietinum,species,2759,Eukaryota,33090,Viridiplantae
ENFVYIAK,3847,Glycine max,species,2759,Eukaryota,33090,Viridiplantae,,,,
...
```

This output corresponds to the tree structure that appears at the left of Figure 3.4 or the tree drawn in the Lineage tree tab on the page that shows the results of a Tryptic Peptide Analysis in the Unipept web interface. Note that the tryptic peptide ENFVYIAK was only found in a peach protein (*Prunus persica*), whereas its `I=L` variant was found in proteins of a species of wild banana (*Musa acuminata* subsp. *malaccensis*) and in different members of the flowering plants including chick pea (*Cicer arietinum*), broad bean (*Vicia faba*), soybean (*Glycine max*), common bean (*Phaseolus vulgaris*), barrel medic (*Medicago truncatula*), orange (*Citrus sinensis*), clementine (*Citrus clementina*) and common grape vine (*Vitis vinifera*).

The Unipept implementation of the LCA algorithm can be applied on a given tryptic peptide using the `unipept pept2lca` command. Using the `-e` option will again have an influence on the LCA computation for the tryptic peptide ENFVYIAK. After all, the LCA will be computed for all taxa associated with proteins in which the tryptic peptide (or one of its `I=L` variants) was found.

```
$ unipept pept2lca ENFVYIAK
peptide,taxon_id,taxon_name,taxon_rank
ENFVYIAK,35493,Streptophyta,phylum
```

```
$ unipept pept2lca ENFVYLAK
peptide,taxon_id,taxon_name,taxon_rank
ENFVYLAK,3760,Prunus persica,species
$ unipept pept2lca -e ENFVYLAK
peptide,taxon_id,taxon_name,taxon_rank
ENFVYLAK,35493,Streptophyta,phylum
```

The correctness of the computed LCAs can be checked based on the taxonomic hierarchy shown in Figure 3.4.

### 3.2.3 Case study: analysis of a metaproteomics data set

As a demonstration of the Unipept CLI, this second case study shows how it can be used to get insight into the biodiversity within one of the faecal samples from a gut microbiome study (Verberkmoes et al., 2009). The sample was taken from a female that is part of a healthy monozygotic twin pair born in 1951 that was invited to take part in a larger double-blinded study. Details of this individual with respect to diet, antibiotic usage, and so on are described by Dicksved et al. (2008) (individual 6a in this study, sample 7 in the study of Verberkmoes et al. (2009)). The most important thing that we learn from the available information in the questionnaire that this individual has filled up, is that she had gastroenteritis at the time the sample was taken and that her twin sister (individual 6b in the study of Dicksved et al. (2008), sample 7 in the study of Verberkmoes et al. (2009)) had taken non-steroidal anti-inflammatory drugs during the past 12 months before the time of sampling. The data can be downloaded from the website of the study and is also available as a demo data set on the Unipept website.

### Duplicate peptides

Say that we stored the list of tryptic peptides that were extracted from sample 7 in the study of Verberkmoes et al. (2009) in the text file `sample7.dat`. The file contains a list of all tryptic peptides, each on a separate line. The following session shows that this file contains a list of 3 983 tryptic peptides (2 065 unique peptides) that could be identified in the faecal sample using *shotgun metaproteomics*.

```
$ head -n5 sample7.dat
SGIVLPGQAQEKPQQAEVVAVGPGGVVDGK
SGIVLPGQAQEKPQQAEVVAVGPGGVVDGK
SGIVLPGQAQEKPQQAEVVAVGPGGVVDGKEVK
MEVAVGDKVIYSK
MDGTEYIIVK
$ wc -l sample7.dat
3983 sample7.dat
$ sort -u sample7.dat | wc -l
2065
```

The first thing that strikes the eye is that a mass spectrometer might pick up multiple copies of the same tryptic peptide from an environmental sample. Depending on the fact whether or not we can draw quantitative conclusion on the number of different identifications of a particular peptide (apart from identification, the quantification of proteins in an environmental sample is an important research theme (Seifert et al., 2013; Kolmeder and Vos, 2014)), we might decide to deduplicate the peptides before they are analyzed further using the Unipept CLI. This decision has an impact on the analysis results, but deduplication also results in improved performance since it avoids duplicate work.

### Non-tryptic peptides

What might be less obvious at first sight, is that the peptides on lines 3 and 4 in the text file `sample7.dat` actually aren't tryptic peptides, but the composition of two tryptic peptides. This is a consequence of the fact that cleavage of proteins using trypsin is not always perfect, leading to some proteins that aren't cleaved properly. Such composed tryptic peptides are called *missed cleavages*. The index structure underpinning Unipept only indexes tryptic peptides that result from an *in silico* trypsin digest of the proteins in UniProt, so that missed cleavages cannot be matched directly by Unipept.

To cope with this problem, we can start to check if the peptides resulting from a shotgun metaproteomics experiment need to be cleaved further before making taxonomic identifications using Unipept. Performing an *in silico* trypsin digest can be done using the `prot2pept` command from the Unipept CLI. This command is executed purely client side, and thus is provided as a standalone command and not as a subcommand of the `unipept` command.

```
$ sed -ne '4{p;q}' sample7.dat
MEVAVGDKVIYSK
$ sed -ne '4{p;q}' sample7.dat | prot2pept
MEVAVGDK
VIYSK
```

### Lowest common ancestor

Once a peptide is broken into multiple tryptic peptides, the lowest common ancestor can be computed for each tryptic peptide using the `unipept pept2lca` command. Next to accepting tryptic peptides as arguments, the command can also read one ore more tryptic peptides from standard input if no arguments were passed. Each tryptic peptide should be on a separate line when using standard input.

```
$ unipept pept2lca -e SGIVLPGQAQEKPQQAEVVAVGPGGVVDGK MDGTEYIIVK
peptide,taxon_id,taxon_name,taxon_rank
SGIVLPGQAQEKPQQAEVVAVGPGGVVDGK,1263,Ruminococcus,genus
MDGTEYIIVK,1263,Ruminococcus,genus
$ sed -ne '3{p;q}' sample7.dat
```

```
SGIVLPGQAQEKPQQAEVVAVGPGGVVDGKEVK
$ sed -ne '3{p;q}' sample7.dat | unipept pept2lca -e
$ sed -ne '3{p;q}' sample7.dat | prot2pept
SGIVLPGQAQEKPQQAEVVAVGPGGVVDGK
EVK
$ sed -ne '3{p;q}' sample7.dat | prot2pept | unipept pept2lca -e
peptide,taxon_id,taxon_name,taxon_rank
SGIVLPGQAQEKPQQAEVVAVGPGGVVDGK,1263,Ruminococcus,genus
```

Unipept only indexes tryptic peptides extracted from UniProt sequences that have a length between 5 and 50 amino acids (boundaries included). This choice was driven by the detection limits of most common mass spectrometers. As a result, an additional time saver is to search for tryptic peptides that have less than 5 of more than 50 amino acids, because Unipept will never find protein matches for these peptides. The `peptfilter` command from the Unipept CLI can be used to filter out peptides that are too short or too long prior to the taxonomic identification step. By default, it filters out all peptides for which it is known in advance that Unipept will find no matches.

```
$ sed -ne '3{p;q}' sample7.dat | prot2pept | peptfilter | \
  unipept pept2lca -e
peptide,taxon_id,taxon_name,taxon_rank
SGIVLPGQAQEKPQQAEVVAVGPGGVVDGK,1263,Ruminococcus,genus
```

All commands of the Unipept CLI follow the input/output paradigm of the Unix command line, so that they be chained together seamlessly. This way, for example, we can determine the LCAs for the first six peptides of sample 7 by combining the previous processing steps: split missed cleavages, filter out peptides that are too short or too long, equate leucine (residue `L`) and isoleucine (residue `I`), and deduplicate the tryptic peptides.

```
$ head -n6 sample7.dat | prot2pept | peptfilter | tr I L | sort -u
GLTAALEAADAMTK
MDGTEYLLVK
```

```
MEVAVGDK
SGLVLPGQAQEKPQQAEVVAVGPGGVVDGK
VLYSK
$ head -n6 sample7.dat | prot2pept | peptfilter | tr I L | sort -u | \
  unipept pept2lca -e
peptide,taxon_id,taxon_name,taxon_rank
GLTAALEAADAMTK,186802,Clostridiales,order
MDGTEYLLVK,1263,Ruminococcus,genus
MEVAVGDK,1263,Ruminococcus,genus
SGLVLPGQAQEKPQQAEVVAVGPGGVVDGK,1263,Ruminococcus,genus
VLYSK,1,root,no rank
```

### Website comparison

The biodiversity in sample 7 from the study of Verberkmoes et al. (2009) can be easily computed and visualized using the Metagenomics Analysis feature of the Unipept web site. All it takes is to paste the list of peptides that were identified from an environmental sample in a text area, select the appropriate search options, and to click the Search button to launch the identification process.

In the session that is shown in Figure 3.6, we have indicated that no distinction should be made between leucine (`L`) and isoleucine (`I`), that the peptides must be deduplicate prior to the actual biodiversity analysis, and that the results must be exported in csv format (comma separated values). Breaking up the missed cleavages happens by default. In addition, the option Advanced missed cleavage handling can be activated to indicate that the results should be aggregated as a post-processing step (not selected in this example).

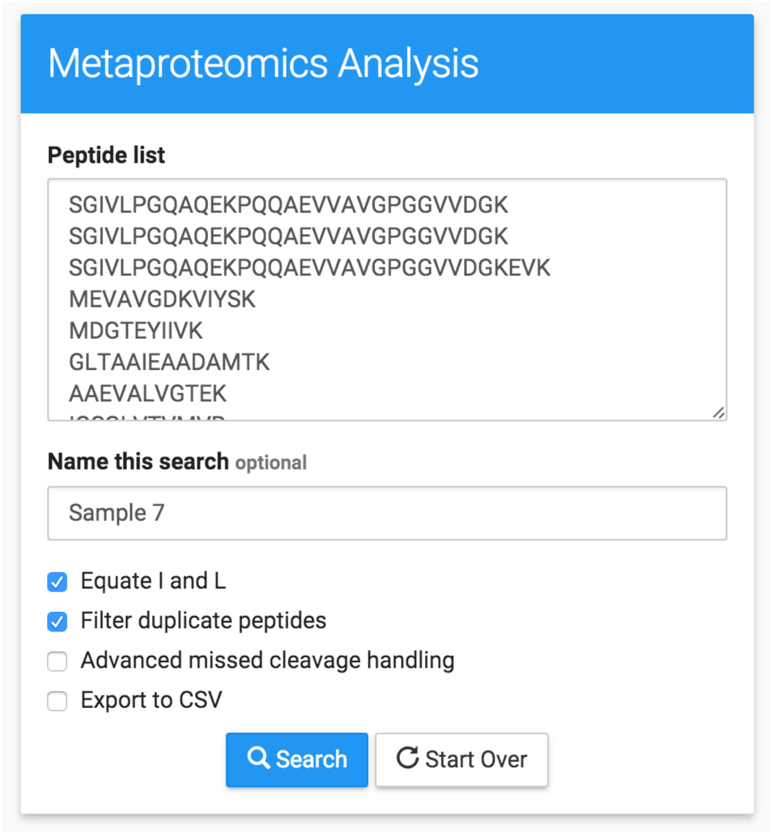
The same result can be obtained using the following combination of commands from the Unipept CLI. The timing gives an impression of the performance of Unipept to compute the LCAs for all 2005 unique tryptic peptides extracted from sample 7. It indicates that part of the processing is parallelized, and that the majority of the processing time is consumed by exchanging data between the client and the Unipept server and the server-side processing of the data.

```
$ prot2pept < sample7.dat | peptfilter | tr I L | sort -u | wc -l
2005
$ time prot2pept < sample7.dat | peptfilter | tr I L | sort -u | \
  unipept pept2lca -e > sample7.csv

real    0m0.329s
user    0m0.465s
sys     0m0.038s
$ head -n6 sample7.csv
peptide,taxon_id,taxon_name,taxon_rank
AAALNLVPNSTGAAK,2,Bacteria,superkingdom
AAALNTLAHSTGAAK,1678,Bifidobacterium,genus
AAALNTLPHSTGAAK,1678,Bifidobacterium,genus
AAAMSMLPTSTGAAK,2,Bacteria,superkingdom
AAANESFGYNEDELVSSDLVGMR,186802,Clostridiales,order
```

For those that are not familiar with IO redirection, the `unipept` command also supports the `-i/--input` option to read the peptides from the file that is passed as an argument and the `-o/--output` option to store the results in a file that is passed as an argument.

```
$ unipept pept2lca --input sample7.dat --output sample7.csv
$ head -n6 sample7.csv
peptide,taxon_id,taxon_name,taxon_rank
AAALNLVPNSTGAAK,2,Bacteria,superkingdom
AAALNTLAHSTGAAK,1678,Bifidobacterium,genus
AAALNTLPHSTGAAK,1678,Bifidobacterium,genus
AAAMSMLPTSTGAAK,2,Bacteria,superkingdom
AAANESFGYNEDELVSSDLVGMR,186802,Clostridiales,order
```


Metaproteomics Analysis

Peptide list

```
SGIVLPGQAQEKPQQAEVVAVGPGGVVDGK
SGIVLPGQAQEKPQQAEVVAVGPGGVVDGK
SGIVLPGQAQEKPQQAEVVAVGPGGVVDGKEVK
MEVAVGDKVIYSK
MDGTEYIIVK
GLTAAIEAADAMTK
AAEVALVGTEK
```

Name this search optional: Sample 7

- [✓] Equate I and L
- [✓] Filter duplicate peptides
- [ ] Advanced missed cleavage handling
- [ ] Export to CSV

**Figure 3.6** Processing of sample 7 from the study of Verberkmoes et al. (2009) using the Metaproteomics Analysis feature of the Unipept web site.

If needed, the unipept `pept2lca` command can be used in combination with the `-a` option to fetch the complete lineages for all LCAs according to the Unipept Taxonomy. Figure 3.7 shows the hierarchical classification of the taxa that could be identified in sample 7. A similar tree view can be found in the *Treeview* tab on the page showing the results of a Metaproteomics analysis in the Unipept web interface.

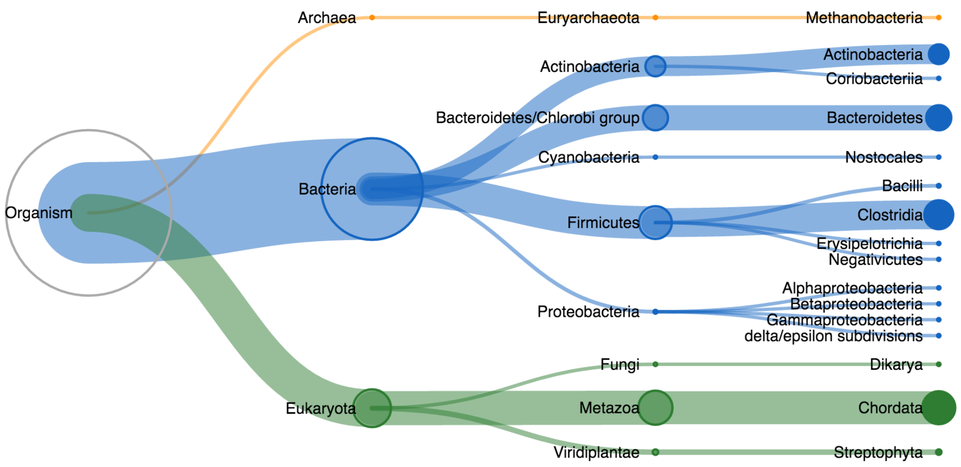


**Figure 3.7** Screenshot of an interactive tree view that shows the results of the biodiversity analysis of sample 7, a metaproteomics data set from the study of Verberkmoes et al. (2009).

# Chapter 4

# Peptidome analysis

This chapter describes the Unique Peptide Finder, a tool for finding taxon-specific peptides that can be used as biomarkers for targeted proteomics. Together with the Peptidome Similarity feature, a tool for comparing the similarity of proteomes, the Unique Peptide Finder forms the Peptidome Analysis part of the Unipept web application.

## 4.1 The unique peptidome: taxon-specific tryptic peptides as biomarkers for targeted metaproteomics

This section contains a verbatim copy of the research article by Mesuere et al. (2016a) as submitted to Proteomics.

**Abstract** — The unique peptide finder (http://unipept.ugent.be/peptidefinder) is an interactive web application to quickly hunt for tryptic peptides that are unique to a particular species, genus or any other taxon. Biodiversity within the target taxon is represented by a set of proteomes selected from a monthly updated list of complete and non-redundant UniProt proteomes, supplemented with proprietary proteomes loaded into persistent local browser storage. The software computes and visualizes pan and core peptidomes as unions and intersections of tryptic peptides occurring in the selected proteomes. In addition, it also computes and displays unique peptidomes as the set of all tryptic peptides that occur in all selected proteomes but not in any UniProt record not assigned to the target taxon. As a result, the unique peptides can serve as robust biomarkers for the target taxon, e.g., in targeted metaproteomics studies. Computations are extremely fast since they are underpinned by the Unipept database, the Lowest Common Ancestor algorithm implemented in Unipept and modern web technologies that facilitate in-browser data storage and parallel processing.

### 4.1.1 Introduction

The proteomics field is recently moving into more targeted approaches driven by key technologies such as Multiple Reaction Monitoring (Pan et al., 2012; Huttenhain et al., 2012). The idea is to program triple quadrupole-like instruments to quantitatively determine predefined molecules by setting them to detect precursor ion masses and predefined MS/MS derived fragments. Target peptides can be selected according to a wide range of criteria such as their protein-specificity to screen for the presence/absence of certain proteins in proteomics studies or their taxon-specificity to screen for the presence/absence of certain taxa in metaproteomics studies. With protein profiling providing assays closer to activated functions, such metaproteome-wide association studies have the potential to become an important tool in environmental studies and modern medicine, and could answer major yet unmet environmental and clinical needs (Juste et al., 2014).

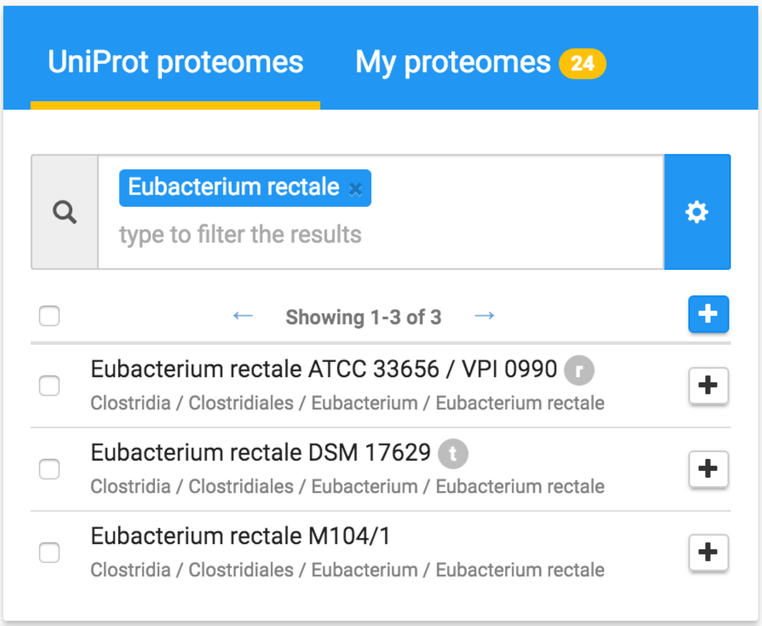
In this paper we present the Unique Peptide Finder, an open source web application that enables the discovery of tryptic peptides that can be used as robust taxon-specific biomarkers. The software package is an integrated component of the Unipept ecosystem (Mesuere et al., 2012; Mesuere et al., 2015) and only takes a few milliseconds to compute all tryptic peptides that are not found in any known protein outside the target taxon and are shared by a sample of proteomes that represent the diversity within the target taxon. This set of tryptic peptides is called the unique peptidome and is visualized in a chart that also displays peptidome sizes and the core and pan peptidomes that give similar insights as the pan and core genomes (Medini et al., 2005) but on a peptide level. Due to the focus on modern web technologies and extremely fast and parallel data processing the chart is highly interactive: including, removing, and rearranging proteomes results in almost instant recalculation of the unique peptidome. Both the data visualizations and the obtained sets of peptidomes can be easily exported for further processing.

## 4.1.2 Web application

The backend of the Unique Peptide Finder is powered by a database that is rebuilt monthly using a completely automated pipeline. It contains all tryptic peptides extracted from the protein records in the UniProt Knowledgebase (Wu et al., 2006), along with taxonomic information extracted from the NCBI Taxonomy database (Wheeler et al., 2004). These data sources are processed according to the procedures described in (Mesuere et al., 2012), resulting in an index of tryptic peptides having a length between 5 and 50 amino acids and precomputed lowest common ancestors that represent their taxonomic distribution. The construction pipeline also preprocesses and integrates the list of all complete and non-redundant proteomes as published on the UniProt website. The resulting database provides quick retrieval of the set of tryptic peptides of all UniProt proteomes and the taxon-specificity of these tryptic peptides.

The web application itself was built using modern web technologies such as JavaScript (ES2015), HTML5, and CSS3. The D3.js library (Bostock, Ogievetsky, and Heer, 2011) was used to implement the interactive visualizations. Fetching sets of tryptic peptides for selected proteomes is the only step in the biomarker analysis that requires communication with the web server that runs the Ruby on Rails backend of Unipept. All other data processing is done client side using JavaScript workers. These workers run as a separate background processes to guarantee that the application remains responsive while the proteomes are being processed. Workers can request data from the database by doing asynchronous requests to the web server. Upon receiving such a request, the web server queries the database and sends the data back to the worker using the JSON (JavaScript Object Notation) format. The combination of these technologies results in a high performance, interactive web application with compelling visualizations without the need to install any software. The only requirement is a modern browser and an Internet connection.

The user interface of the Unique Peptide Finder is dominated by a large chart at the top of the page. Proteomes can be included, removed or rearranged using control elements underneath the chart or by direct interaction with the chart itself. The "Analyzed proteomes" at the right-hand side lists all proteomes currently included in the analysis. The "Proteome library" at the left-hand side lists all available proteomes: complete and non-redundant proteomes published by UniProt ("UniProt proteomes") and proteomes loaded into persistent local browser storage ("My proteomes"). The "UniProt proteomes" library currently contains over 10 000 proteomes and is updated monthly with each new UniProt release. Additional proteomes can be added to the "My proteomes" library by loading multi-fasta files that each contain the protein sequences of a single proteome. Processed file contents are persistently stored in the browser so that they can be reused across intermittent sessions and are never stored on the Unipept server so that confidentiality of proprietary proteomes is never compromised. Proteomes are listed on pages of 50 hits and can be filtered on taxonomic rank, type strain status (Federhen et al., 2014), reference status (as assigned by UniProt) and name (Figure 4.1). Plus-buttons can be clicked to include all or individual proteomes to the analysis.



Eubacterium rectale ATCC 33656 / VPI 0990 r
Clostridia / Clostridiales / Eubacterium / Eubacterium rectale

Eubacterium rectale DSM 17629 t
Clostridia / Clostridiales / Eubacterium / Eubacterium rectale

Eubacterium rectale M104/1
Clostridia / Clostridiales / Eubacterium / Eubacterium rectale

**Figure 4.1** Screenshot of the proteome library with the UniProt proteomes filtered on *Eubacterium rectale*. The first hit is marked as a reference proteome by UniProt. The second hit is marked as a type strain by NCBI. Taxonomic breadcrumbs show class, order, genus, and species to which the proteomes were assigned. Clicking any of the breadcrumbs activates the corresponding taxon as a new taxonomic filter. Plus icons on the right can be clicked to include all or individual proteomes to the analysis.

### 4.1.3 Analysis

Each proteome included in the analysis is converted into its corresponding (tryptic) *peptidome*: the set of all (tryptic) peptides encoded in the proteome of the organism. The name of the first proteome included appears at the leftmost side of the chart's x-axis and a grey data point represents its peptidome size (the number of peptides in its peptidome). When a new proteome gets included in the analysis it appears at the rightmost side of the chart, with three additional data points representing sizes of the pan, core, and unique peptidomes of all proteomes that have been included in the analysis so far. The *pan peptidome* (blue data point) is the union of all these peptidomes and the *core peptidome* (orange data point) is their intersection. Peptides in the core peptidome occur in all proteomes included so far, but might also occur in proteomes of other taxa. To be useful as biomarkers for targeted metaproteomics, the core peptidome is further reduced by removing all peptides occurring in UniProt records that are not assigned to the target taxon. The target taxon is automatically computed as the lowest common taxonomic ancestor of all proteomes included in the analysis, using the Lowest Common Ancestor (LCA) algorithm implemented in Unipept (Mesuere et al., 2012). This reduced set is called the *unique peptidome* and is represented by a green data point.

All unions and intersections are computed solely from data already available in the browser. Determining unique peptidomes is more compute-intensive and requires knowledge about the taxon-specificity of tryptic peptides that is only available in the backend database. Their computations can be speeded up by carefully avoiding unnecessary computations and relying on precomputed lowest common taxonomic ancestors for all tryptic peptides in the Unipept database. Exact sizes of peptidomes are shown in a tooltip that appears when hovering over the data points in the chart (Figure 4.2). Clicking a data point displays a popup that provides options to remove the corresponding proteome from the analysis or download any of the four lists of peptides related to that proteome. This allows further data analysis or filtering of the unique peptidome (e.g., based on sequence length, peptide sequence, presence of amino acids that are prone to chemical modification, etc.) using specialized software for targeted (meta)proteomics such as Skyline (MacLean et al., 2010).

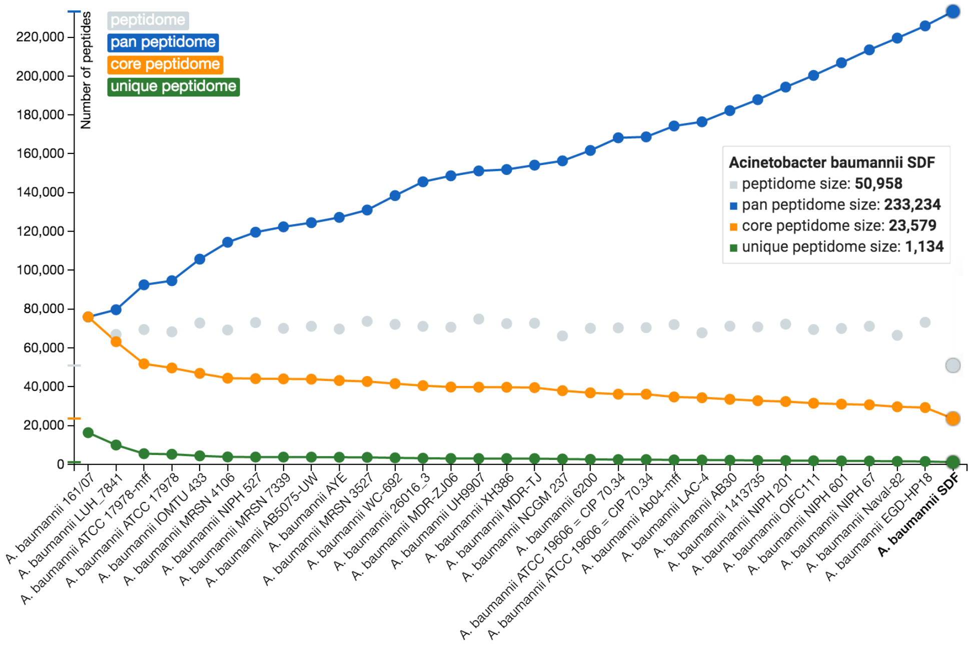


**Figure 4.2** Chart displaying sizes of peptidomes (gray), pan peptidomes (blue), core peptidomes (orange) and unique peptidomes (green) for 32 *Acinetobacter baumannii* proteomes available in the UniProt proteomes library. The core peptidome of *A. baumannii* stabilizes gradually with 23 579 tryptic peptides occurring in all proteomes included in the analysis. Of these common peptides, 1 134 peptides in the unique peptidome are found to be specific for *A. baumannii*. The smaller peptidome size of *A. baumannii SDF* is no error in the application or reference database, but the result of a genome that is 20% reduced (Fournier et al., 2006).

Since pan, core, and unique peptidomes of a particular proteome are always computed based on the proteomes to its left in the chart, the order of proteomes included in the analysis matters. Proteomes can be easily rearranged by simply dragging them to the desired location in the chart or the “Analyzed proteomes” table. The latter also provides a couple of automatic rearrangement heuristics (e.g., alphabetically, on peptidome size, etc.). Removing proteomes is equally easy by dragging proteomes to the trash can on the chart, or by clicking the corresponding trash icon in the “Analyzed proteomes” table. Pan, core, and unique peptidomes that are affected after inclusions, removals or rearrangements are instantly recalculated and size updates are animated in the chart. The chart supports fullscreen mode that also provides all manipulations needed for biomarker analysis.

### 4.1.4 Discussion

The rightmost unique peptidome in the chart theoretically represents robust taxon-specific biomarkers for metaproteomics analysis, since it contains tryptic peptides that occur in all the proteomes included in the analysis and not in a single UniProt record that is not assigned to the target taxon. In practice, however, sensitivity of the biomarkers depends on completeness of the included proteomes and how well they sample diversity within the target taxon. Their specificity depends on how well the global protein space has been sampled in the UniProt database. In addition, the analysis also depends on the correctness of the taxonomic annotations of the proteomes and UniProt entries. The value of the Unique Peptide Finder and some potential pitfalls are further discussed in the context of two case studies.

In a first case study we determined the unique peptidome of *Acinetobacter baumannii*. For this bacterial species, all 32 proteomes available in the "UniProt proteomes" library were included in the analysis. The proteome *A. baumannii 161/07* (UniProt Proteome UP000034216) with the largest peptidome size was chosen as the leftmost peptidome in the chart and the other proteomes were automatically sorted using the "optimize pan and core peptidome" heuristic (Figure 4.2). The resulting chart displays a steadily increasing pan peptidome (blue line), with an average of around 5 000 proteome-specific peptides and an average peptidome size of around 70 000. The core peptidome (orange line) levels off after a handful of proteomes, with a total of

23 579 tryptic peptides shared by all included proteomes. Of these shared peptides 1 134 do not occur in any species other than *A. baumannii*. As this unique peptidome (green line) contains peptides that are both common and unique to *A. baumannii*, it can be used to detect the occurrence of the species in metaproteome data sets.

The fact that biomarker analysis always relies on correct taxonomic assignments is illustrated by the misclassified proteome *A. baumannii 6411* (UniProt Proteome UP-000031110) that was deliberately excluded from the above analysis. Using the Peptidome Clustering application (another tool from the Unipept ecosystem that is available online, unpublished) to cluster this proteome with available type strain proteomes of the genus *Acinetobacter*, shows that the proteome correctly classifies in the species *Acinetobacter nosocomialis*. As a result of this misclassification the Unique Peptide Finder discovers almost no unique peptides for the species for *A. nosocomialis*. Most of its actual unique peptides also occur in the proteome that was incorrectly classified as *A. baumannii*, so that the tool does not consider them to be specific to *A. nosocomialis*. The upside is that the Unique Peptide Finder in combination with the Peptidome Clustering tool allows to easily spot such misclassifications, after which they can be reported to curators of the underpinning information sources (in this case UniProt).

In a second case study we looked at potential biomarkers for *Bacillus anthracis*, a bacterial species that is hard to identify because of its relatedness with *Bacillus cereus*. For this species, the 3 proteomes available in the "UniProt proteomes" library were included in the analysis, together with proteomes of 24 other *B. anthracis* assemblies that were downloaded from the NCBI website (Table 4.1) and loaded into the "My proteomes" library. The resulting chart (Figure 4.3) displays a core peptidome containing 50 724 peptides, of which 878 are specific to *B. anthracis*. These unique peptides can be used to discern *B. anthracis* from *B. cereus* and any other species. When we revisit the 11 *B. anthracis* biomarkers proposed by Chenau et al. (2014), we first observe that only 8 of them are true tryptic peptides. These 8 peptides are indeed found in *B. anthracis* proteomes and not in any *Bacillus cereus* proteome. However, 3 of these peptides are also found in non-*Bacillus cereus* proteomes: LVGGVAVIK in a range of other bacteria, ILDQSADK in several Firmicutes and VCTITGR in a *Turicibacter sanguinis* proteome. This can be easily verified using the Tryptic Peptide Analysis tool in Unipept. Only one of the 8 tryptic peptides (WLLRPEDPNYVLIK) was common to all 27 *B. anthracis* proteomes we analyzed in this case study.

## 4.1.5 Conclusion

The Unique Peptide Finder is a user-friendly and high performance web application to find robust taxon-specific biomarkers for use in targeted metaproteomics studies. Biomarker analysis can combine a selection of UniProt proteomes and proprietary proteomes that are persistently stored in the browser. The tool is built using modern web technologies that provide parallel non-blocking computations, interactive visualizations and export functionality.

**Table 4.1** Overview of the *Bacillus anthracis* assemblies used to analyze the *B. anthracis* peptidome. The order of the assemblies corresponds to the order of the proteomes in Figure 4.3

| ASSEMBLY NAME | ASSEMBLY ACCESSION NUMBER |
|---|---|
| Bacillus anthracis 52-G | GCA_000559005.1 |
| Bacillus anthracis CZC5 | GCA_000534935.1 |
| Bacillus anthracis (strain CDC 684 / NRRL 3495) | GCA_000021445.1 |
| Bacillus anthracis str. Vollum | GCA_000742895.1 |
| Bacillus anthracis str. Turkey32 | GCA_000833275.1 |
| Bacillus anthracis str. Sterne | GCA_000832635.1 |
| Bacillus anthracis str. V770-NP-1R | GCA_000832785.1 |
| Bacillus anthracis 9080-G | GCA_000558985.1 |
| Bacillus anthracis 8903-G | GCA_000558965.1 |
| Bacillus anthracis (strain A0248) | GCA_000022865.1 |
| Bacillus anthracis str. Carbosap | GCA_000732465.1 |
| Bacillus anthracis str. A16 | GCA_000512835.1 |
| Bacillus anthracis str. SVA11 | GCA_000583105.1 |
| Bacillus anthracis Tsiankovskii-I | GCA_000181675.2 |
| Bacillus anthracis str. A0174 | GCA_000182055.1 |
| Bacillus anthracis str. A0193 | GCA_000181915.1 |
| Bacillus anthracis str. A0389 | GCA_000219895.1 |
| Bacillus anthracis str. A0488 | GCA_000181835.1 |
| Bacillus anthracis str. A0465 | GCA_000181995.1 |
| Bacillus anthracis str. A0442 | GCA_000181935.1 |
| Bacillus anthracis str. BF1 | GCA_000295695.1 |
| Bacillus anthracis str. A16R | GCA_000512775.1 |
| Bacillus anthracis str. UR-1 | GCA_000292565.1 |
| Bacillus anthracis str. 95014 | GCA_000585275.1 |

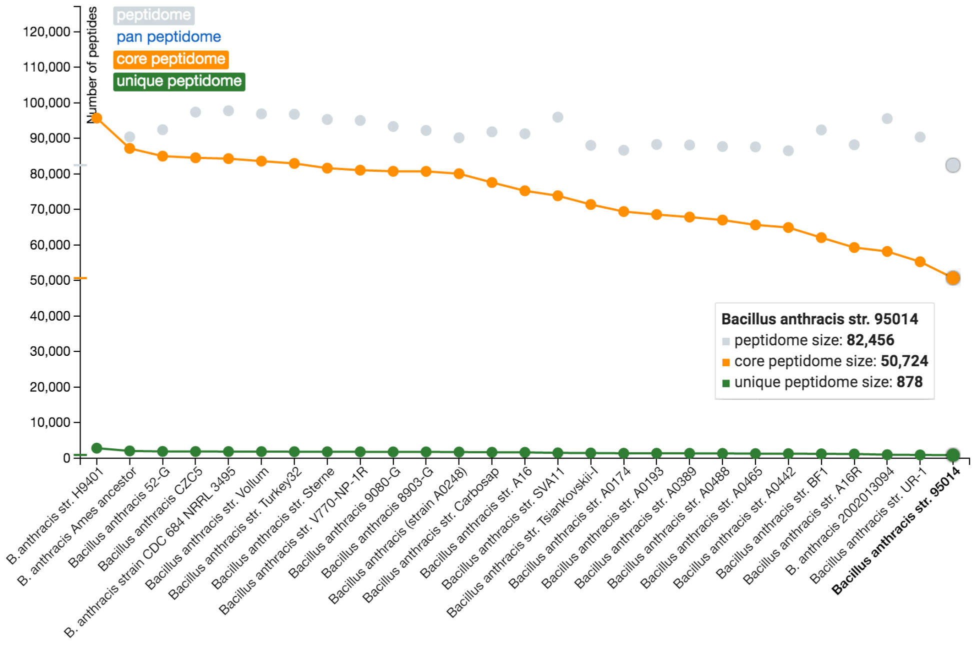


**Figure 4.3** Chart displaying sizes of peptidomes (gray), core peptidomes (orange) and unique peptidomes (green) for 3 *B. anthracis* proteomes available in the "UniProt proteomes" library and 24 *B. anthracis* proteomes downloaded from the NCBI website. These 27 proteomes have 50 724 tryptic peptides in common (core peptidome), of which 878 are specific to *B. anthracis* (unique peptidome). Sizes of the pan peptidomes were hidden in the visualization to improve legibility of the chart.

## 4.2 Peptidome clustering

Using the foundations of the Unique Peptide Finder (Section 4.1), the Peptidome Clustering feature allows users to calculate and visualize a similarity matrix for a set of proteomes. The proteomes, both the ones from UniProt as well as custom proteomes, can be selected using the same user interface as the Unique Peptide Finder. The similarity between two proteomes is determined by looking at the number of peptides they share in their respective peptidomes. Additionally, the hierarchical clustering method UPGMA is used to cluster the proteomes and generate a phylogenet-

UPGMA stands for Unweighted Pair Group Method with Arithmetic Mean.

ic tree that is aligned with the similarity matrix (Figure 4.4). Each of the bifurcations of the phylogenetic tree contains a small circular button that enables users to swap the position of the two branches as well as the corresponding rows and columns of the similarity matrix. As with all Unipept visualizations, the image can be easily exported to both png and svg formats. The source data of the similarity matrix can be exported as a csv file containing all similarity values and the phylogenetic tree can be exported in the Newick tree format.

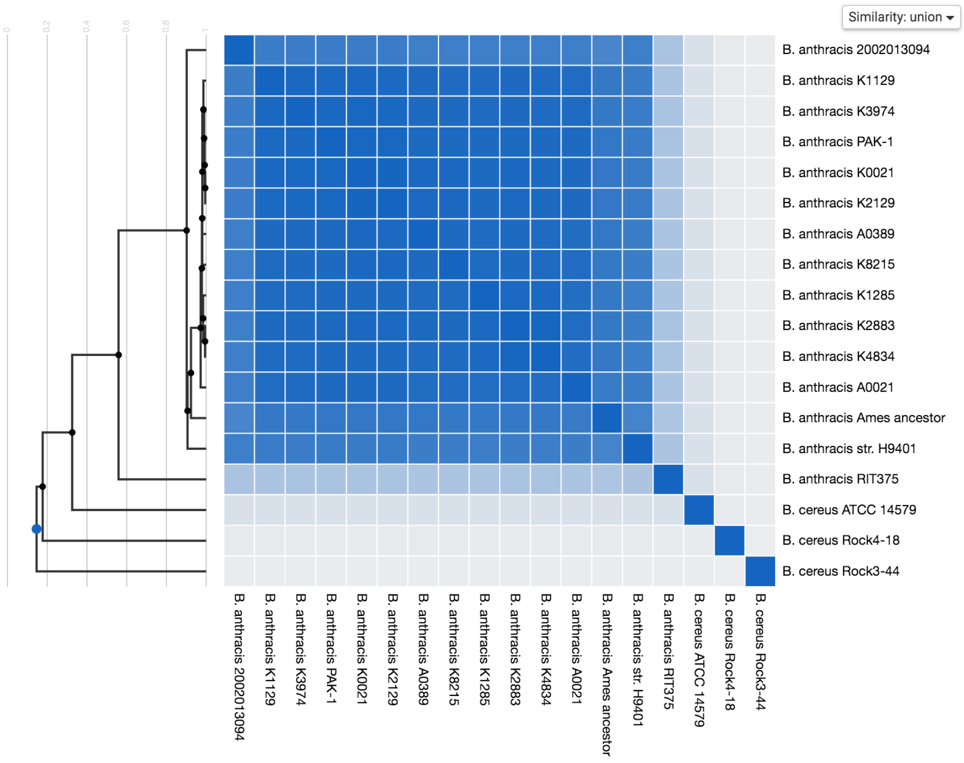


**Figure 4.4** Peptidome clustering of all 15 *Bacillus anthracis* proteomes and the 4 *Bacillus cereus* reference proteomes of UniProt using the "union" similarity measure. The species *B. anthracis* and *B. cereus* are considered very related and hard to distinguish, but the Unipept Peptidome Clustering shows a clear distinction between them.

## Similarity measures

The similarity measure used by the Peptidome Clustering is based on the ratio of peptides that two proteomes have in common. The nominator of this fraction is thus the number of peptides that both have in common, or the size of the intersection of the two sets of peptides. For the value of the denominator, the application offers five options: "Union" is the default option and uses the size of the union of the two sets. This is the recommended option for comparing multiple strains of a single species or closely related species and will always result in the lowest similarity values of the available options. The "Minimum" setting uses the minimum of the size of the two sets and will result in the highest similarity of the five options. This setting is recommended for comparing separate species. The three other options, "Maximum", "Average", and "Ochiai" will result in similarity values that fall between the previous two and will respectively use the maximum, average or square root of the product of the size of the two peptide sets.

## Performance

All similarity calculations are done client-side in the web browser of the user, so the entire peptidome (i.e., the set of tryptic peptides) of each of the proteomes in the analysis needs to be downloaded. Downloading the actual sequences would take too much time, so another solution was needed. Since we only want to know if a peptide is present in two sets, the actual sequences are not needed, only something to uniquely identify them. These identifiers were found in the form of the primary keys used in the underlying Unipept database. Using these id's has two advantages: they are integers and thus smaller than the sequences themselves and they are faster to retrieve from the database. Using integers also allowed us to perform an additional encoding step know as delta encoding to further reduce the

Delta encoding is explained in more detail in Section 5.2.11.

size. Delta encoding transmits the differences between successive values instead of the values themselves. Because our list of integers is sorted, this encoding step results in smaller values and thus a smaller download size. Finally, all data is compressed by the web server using gzip. Gzip is a compression algorithm that is widely supported and built-in into most browsers. The combination of these processing steps allowed us to reduce the download size significantly. For a typical *Escherichia coli* genome, the size was reduced from 1.4 MB for the original peptide sequences to only 63 KB, a spectacular reduction of over 20 times.

To efficiently calculate similarities, we need to be able to quickly compute the union and intersection of sets. Unfortunately, JavaScript has no reliable implementation of the set data type. We had to find an alternative, but sadly none of the set implementations we tried met our performance requirements. Eventually, we used a different strategy and went with simple arrays of the integer identifiers described in the previous paragraph. When these arrays contain sorted integers, it is possible to efficiently calculate the union or intersection of them by simply synchronously iterating over them a single time. Because we don't need the union and intersection itself but only their sizes, we can deduce both sizes by only calculating one of the two and combining it with the already known sizes of the peptidomes. Exploiting the symmetry of the similarity matrix further reduces the computation time by half to under a millisecond per similarity. Calculating the entire similarity matrix for 33 *Acinetobacter baumannii* genomes takes around 180 milliseconds.

# Chapter 5

# A short history of Unipept

This chapter tells the story of Unipept from a developer's point of view. The first section deals with Unipept before it was a web application. Section two goes into detail about the individual versions of the Unipept web application. The last section handles the Unipept command line tools.

## 5.1 Before the web application

The first attempt at creating a tool for peptide analysis was a stand-alone application written in Java. The data source for this application was the set of complete bacterial RefSeq genomes. First, all files containing the proteomes were downloaded in the GenBank flat file format. These files were then fed through our data processing pipeline. The processing consisted of parsing the files using BioJava (Prlić et al., 2012)and performing an in-silico trypsin digest on the protein sequences. We then only store peptides that have a length between 5 and 50 amino acids (boundaries included). This choice was driven by the detection limits of most common mass spectrometers.

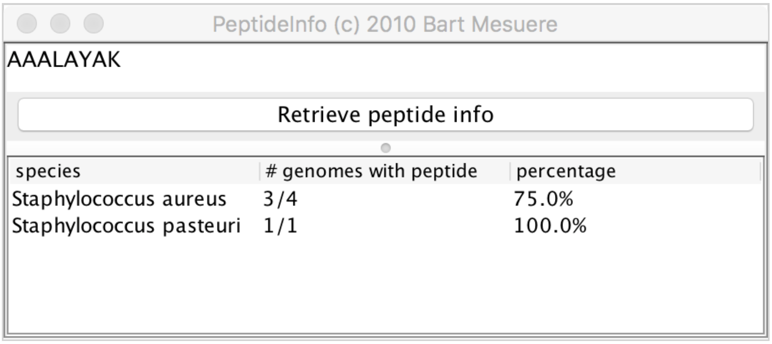
### Storing the data

Unlike in a normal trie, in a patricia trie, parent nodes having only a single child are collapsed into a single node. This reduces the size of the data structure, especially in sparse trees.

Initially, the peptides were stored using a patricia trie. A trie, or prefix tree, is a data structure where an ordered tree is used to efficiently search for keys (in our case tryptic peptides) and retrieve the associated values (in our case taxon id's). While information retrieval is very fast, this approach has two major problems: the entire dataset must fit into memory to achieve optimal performance and it's not straightforward to both reliably and efficiently store the data structure to disk. This effectively means that all source data must be reprocessed every time the program is run.

Both problems were solved by storing the data in a MySQL database instead of a patricia trie. By default, MySQL uses the hard drive to store data and only uses memory for temporary tables and caches. This not only allowed us to process the source data once and have permanent access afterwards, but also enabled more flexible data access through the use of SQL queries. The downsides are slightly slower data access and fairly slower index construction.

At that time, there were 1 190 complete bacterial RefSeq genomes spanning 860 species. Using the MySQL database, it took 21 hours to parse these genomes and create the index. This resulted in a database of 7 GB of which half was used for indexes. The database contained information on 34.4 million distinct peptides that each occurred in 1.57 genomes on average. 90.7% of all peptides only occurred within a single species, 5.8% occurred within two species.



```
species                  # genomes with peptide    percentage
Staphylococcus aureus    3/4                       75.0%
Staphylococcus pasteuri  1/1                       100.0%
```

**Figure 5.1** The initial PeptideInfo Java application, when searching for the peptide AAALAYAK. The peptide was found in 3 of the 4 *Staphylococcus aureus* genomes and in a *Staphylococcus pasteuri* genome. Note that the database only contained a test set of 8 genomes at the moment of the screenshot.

### Exploring the data

After creating a fast index mapping peptides on taxonomic nodes, there are two research questions that emerge: in which species does a given peptide occur, and which peptides only occur within a given species. To answer the first question, a Java application with a graphical user interface was created. As shown in Figure 5.1, the user could enter a tryptic peptide in the text area, click the search button and a report was generated listing all species in which the pep-

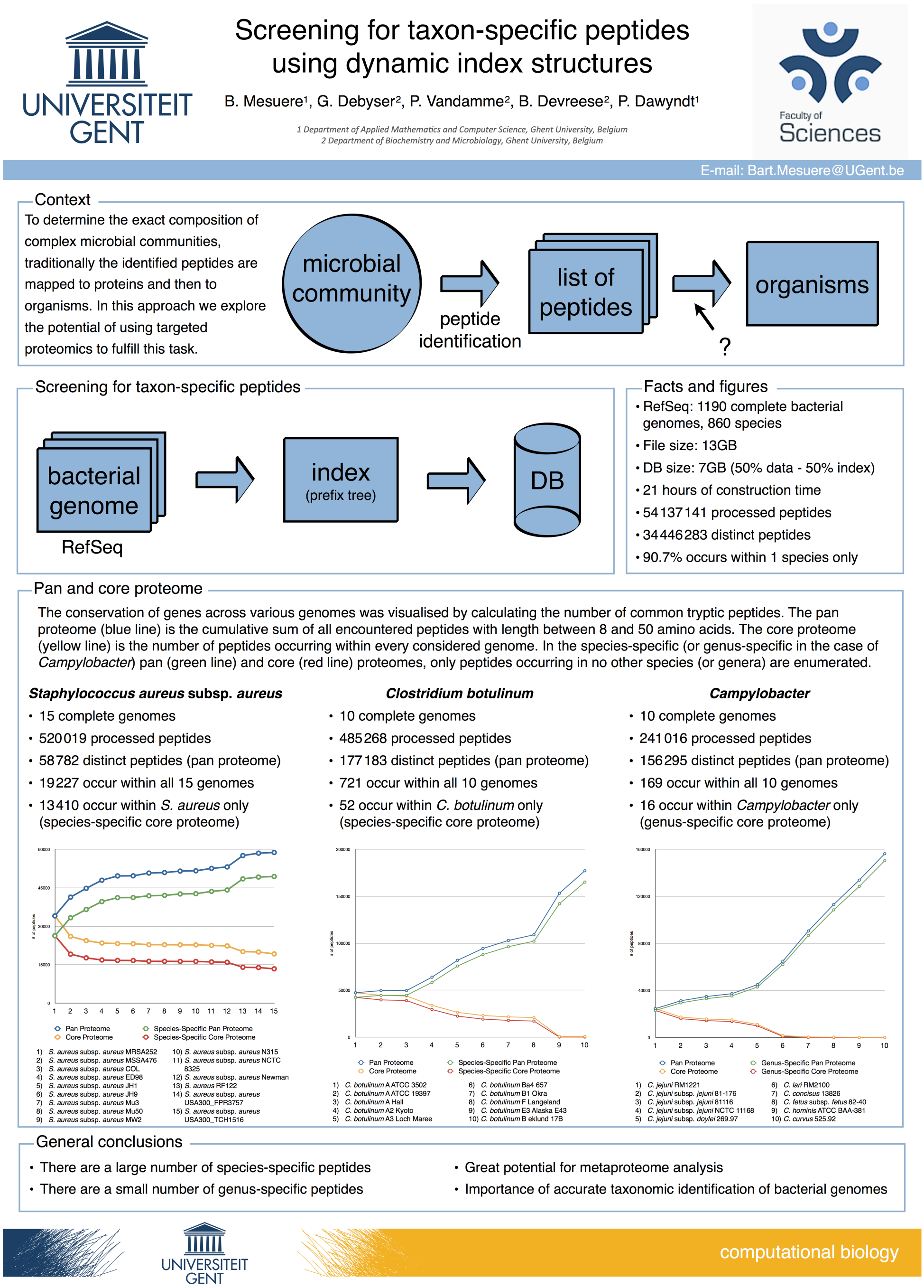
This PeptideInfo application later evolved into the Unipept tryptic peptide analysis.

tide was found. The application also listed the number of genomes in which the peptide was found in case multiple genomes of a species were available.

The second question, which peptides can be used to uniquely identify a certain species, was harder to answer. The database was not optimized for this and the queries took too long to wrap everything into a desktop application. Instead, a collection of scripts and queries was created to explore the potential of the data. Figure 5.2 shows the promising results of such analysis on the available genomes of *Staphylococcus aureus* subsp. *aureus*, *Clostridium botulinum* and *Campylobacter*. These results show that there is a large number of species-specific peptides and a surprisingly low number of genus-specific peptides. This means that there is a great potential to use tryptic peptides as a way to identify organisms.

A solution to this question was later provided by the Unipept unique peptide finder.

Because there was an immediate need for a tool to help analyze the gut microbiota of patients with cystic fibrosis for a joint research project (Debyser et al., 2016), we decided to first further develop the PeptideInfo application.



## Screening for taxon-specific peptides using dynamic index structures

UNIVERSITEIT GENT

B. Mesuere[1], G. Debyser[2], P. Vandamme[2], B. Devreese[2], P. Dawyndt[1]

*1 Department of Applied Mathematics and Computer Science, Ghent University, Belgium*
*2 Department of Biochemistry and Microbiology, Ghent University, Belgium*

Faculty of Sciences

E-mail: Bart.Mesuere@UGent.be

### Context

To determine the exact composition of complex microbial communities, traditionally the identified peptides are mapped to proteins and then to organisms. In this approach we explore the potential of using targeted proteomics to fulfill this task.

microbial community → peptide identification → list of peptides → ? → organisms

### Screening for taxon-specific peptides

bacterial genome → index (prefix tree) → DB

RefSeq

### Facts and figures

- RefSeq: 1190 complete bacterial genomes, 860 species
- File size: 13GB
- DB size: 7GB (50% data - 50% index)
- 21 hours of construction time
- 54 137 141 processed peptides
- 34 446 283 distinct peptides
- 90.7% occurs within 1 species only

### Pan and core proteome

The conservation of genes across various genomes was visualised by calculating the number of common tryptic peptides. The pan proteome (blue line) is the cumulative sum of all encountered peptides with length between 8 and 50 amino acids. The core proteome (yellow line) is the number of peptides occurring within every considered genome. In the species-specific (or genus-specific in the case of *Campylobacter*) pan (green line) and core (red line) proteomes, only peptides occurring in no other species (or genera) are enumerated.

***Staphylococcus aureus* subsp. *aureus***

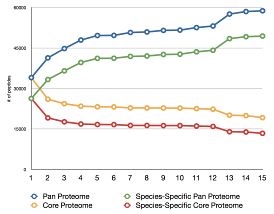
- 15 complete genomes
- 520 019 processed peptides
- 58 782 distinct peptides (pan proteome)
- 19 227 occur within all 15 genomes
- 13 410 occur within *S. aureus* only (species-specific core proteome)

***Clostridium botulinum***

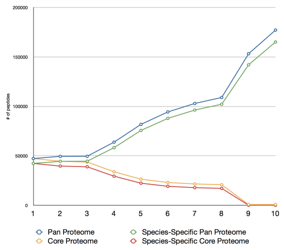
- 10 complete genomes
- 485 268 processed peptides
- 177 183 distinct peptides (pan proteome)
- 721 occur within all 10 genomes
- 52 occur within *C. botulinum* only (species-specific core proteome)

***Campylobacter***

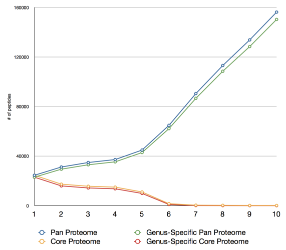
- 10 complete genomes
- 241 016 processed peptides
- 156 295 distinct peptides (pan proteome)
- 169 occur within all 10 genomes
- 16 occur within *Campylobacter* only (genus-specific core proteome)



1) *S. aureus* subsp. *aureus* MRSA252
2) *S. aureus* subsp. *aureus* MSSA476
3) *S. aureus* subsp. *aureus* COL
4) *S. aureus* subsp. *aureus* ED98
5) *S. aureus* subsp. *aureus* JH1
6) *S. aureus* subsp. *aureus* JH9
7) *S. aureus* subsp. *aureus* Mu3
8) *S. aureus* subsp. *aureus* Mu50
9) *S. aureus* subsp. *aureus* MW2
10) *S. aureus* subsp. *aureus* N315
11) *S. aureus* subsp. *aureus* NCTC 8325
12) *S. aureus* subsp. *aureus* Newman
13) *S. aureus* RF122
14) *S. aureus* subsp. *aureus* USA300_FPR3757
15) *S. aureus* subsp. *aureus* USA300_TCH1516



1) *C. botulinum* A ATCC 3502
2) *C. botulinum* A ATCC 19397
3) *C. botulinum* A Hall
4) *C. botulinum* A2 Kyoto
5) *C. botulinum* A3 Loch Maree
6) *C. botulinum* Ba4 657
7) *C. botulinum* B1 Okra
8) *C. botulinum* F Langeland
9) *C. botulinum* E3 Alaska E43
10) *C. botulinum* B eklund 17B



1) *C. jejuni* RM1221
2) *C. jejuni* subsp. *jejuni* 81-176
3) *C. jejuni* subsp. *jejuni* 81116
4) *C. jejuni* subsp. *jejuni* NCTC 11168
5) *C. jejuni* subsp. *doylei* 269.97
6) *C. lari* RM2100
7) *C. concisus* 13826
8) *C. fetus* subsp. *fetus* 82-40
9) *C. hominis* ATCC BAA-381
10) *C. curvus* 525.92

### General conclusions

- There are a large number of species-specific peptides
- There are a small number of genus-specific peptides
- Great potential for metaproteome analysis
- Importance of accurate taxonomic identification of bacterial genomes

UNIVERSITEIT GENT

computational biology

**Figure 5.2** A poster presenting the results of the precursor of Unipept at the 4th International Symposium on Proteome Analysis in Antwerp, Belgium in December 2010.

## 5.2 The Unipept web application

After creating a database-backed mapping from peptide sequences to organisms, the use of Java for the client application was reevaluated. The main advantages were familiarity with the programming language, good performance and, at least theoretically, cross-platformness. This was offset by drawbacks such as the need for installation of the application, the difficulty to distribute updates and the need to run your own database as a user. Because the target audience of Unipept is non-technical users and we had an agile development style in mind, the disadvantages outweighed the benefits.

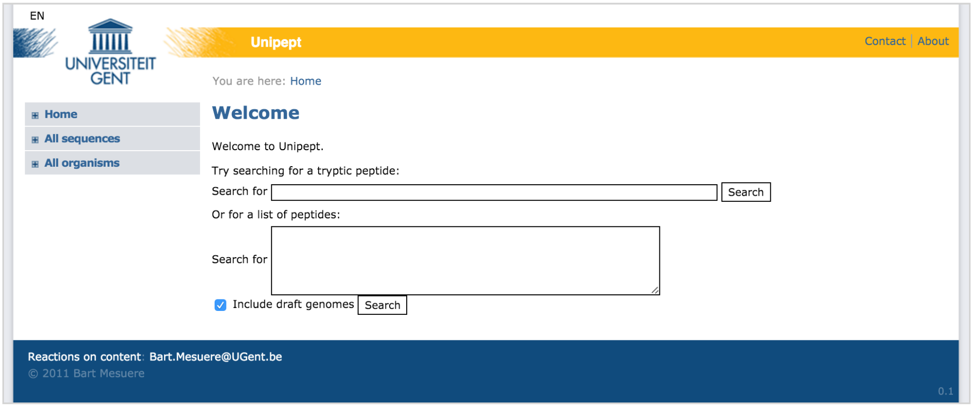
Users would also have to set up a local database, still requiring platform-specific solutions.

### 5.2.1 Unipept version 0.1 – 0.4

**Unipept version 0.1**

Released on February 3, 2011.

In the end, we chose to rebuild the client as a web application using the Ruby on Rails framework (Figure 5.3). The main reason to choose for a web application was the low threshold for users to start using the application and the ease with which new versions can be deployed. This client-server architecture also completely shifts the technical burden away from the user. The choice for Ruby on Rails was a bit of a risk since we had no previous experience with it, but the framework looked promising and was becoming popular very fast.


EN

UNIVERSITEIT GENT

Unipept

Contact | About

You are here: Home

Home
All sequences
All organisms

Welcome

Welcome to Unipept.

Try searching for a tryptic peptide:

Search for [ ] Search

Or for a list of peptides:

Search for [ ]

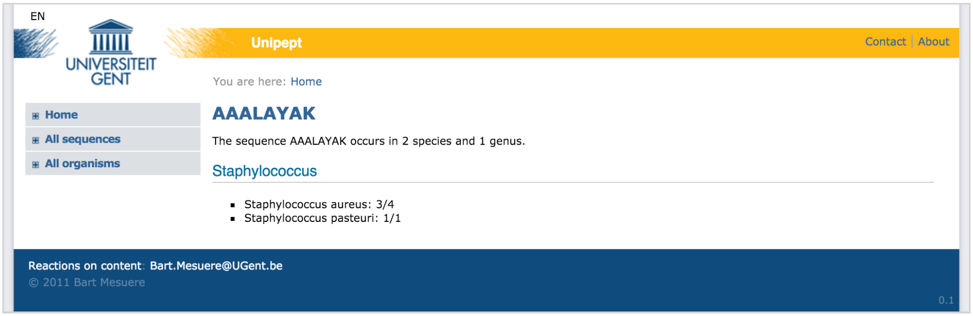
[✓] Include draft genomes Search

Reactions on content: Bart.Mesuere@UGent.be
© 2011 Bart Mesuere

0.1

**Figure 5.3** The homepage of version 0.1 of the Unipept web application.

Unipept version 0.1 was a straightforward reimplementation of the existing PeptideInfo tool. As can be seen in Figure 5.3 and Figure 5.4, the user could submit a tryptic peptide in a search form to which the application responded with an overview in which species the peptide was found. The only noteworthy change to the old application was the inclusion of not only the complete RefSeq genomes, but also the draft genomes.


EN

UNIVERSITEIT GENT

Unipept

Contact | About

You are here: Home

Home
All sequences
All organisms

AAALAYAK

The sequence AAALAYAK occurs in 2 species and 1 genus.

Staphylococcus

- Staphylococcus aureus: 3/4
- Staphylococcus pasteuri: 1/1

Reactions on content: Bart.Mesuere@UGent.be
© 2011 Bart Mesuere

0.1

**Figure 5.4** Web-based reimplementation of the PeptideInfo tool. The analysis for the peptide AAALAYAK is shown.

### Unipept version 0.2

Released on May 26, 2011.

After reaching feature parity with the Java client in version 0.1, work began on adding new features. Where the initial single peptide search only listed the species in which a

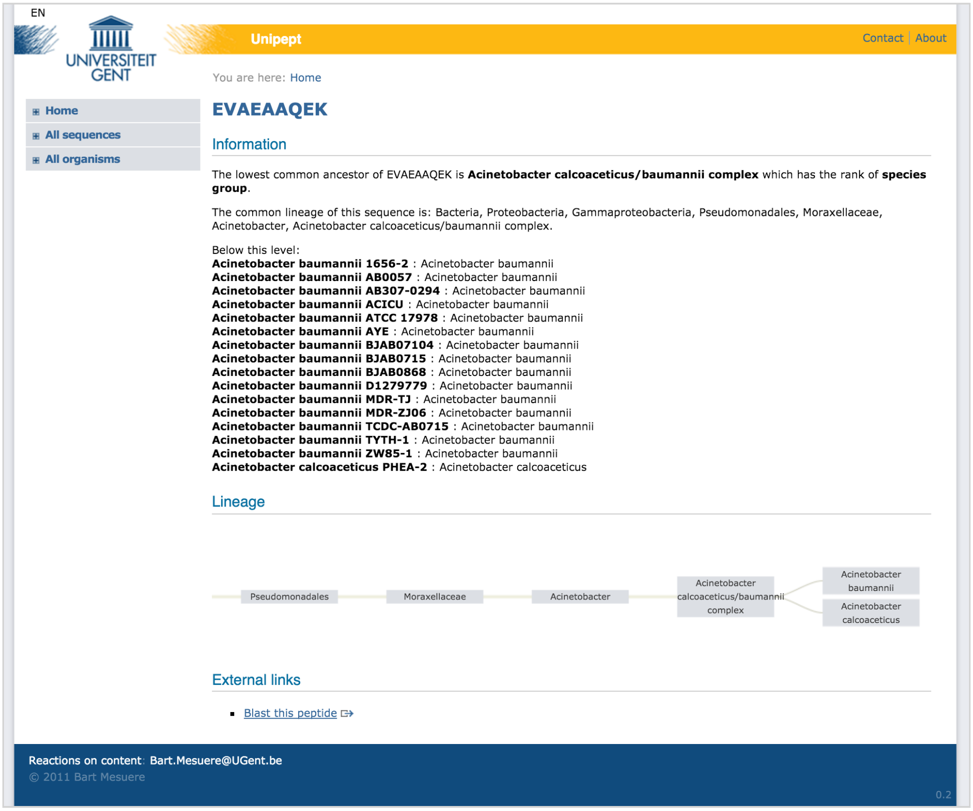
The LCA is discussed in more detail in Section 2.1.2.3.

peptide was found, version 0.2 introduced the concept of the lowest common ancestor (LCA). To efficiently calculate the LCA, the complete lineage of every organism is needed. Until then, the hierarchical information of the taxonomy tree was not easily accessible. Each of the records in the taxonomy table represented a single taxonomic node containing, among other things, the taxon id of its parent. To retrieve all ancestors of a given organism, we needed to recursively query the database for the parent node until we reached the root. A solution to this problem was to calculate the lineage for each organism during database construction, and then store that path to root in a dedicated table. To accommodate for a variable number of ancestors, a fixed structure was used, using only the so-called named ranks. This pre-calculated table containing the 28 possible ancestors for every organism made it possible to efficiently calculate LCAs.

With ancestor, we mean all parent nodes in the taxonomic tree, not evolutionary ancestors.

Unfortunately, it turned out that the unnamed ranks also contain important information, especially in plants.

This lineage data was used to improve the single peptide analysis page (Figure 5.5). Instead of simply listing the species in which the peptide was found, it now shows all organisms, the common lineage of these organisms and the lowest common ancestor. The result is also visualized by drawing a simple treeview of the relevant section of the taxonomy tree.



You are here: Home

EVAEAAQEK

Information

The lowest common ancestor of EVAEAAQEK is **Acinetobacter calcoaceticus/baumannii complex** which has the rank of **species group**.

The common lineage of this sequence is: Bacteria, Proteobacteria, Gammaproteobacteria, Pseudomonadales, Moraxellaceae, Acinetobacter, Acinetobacter calcoaceticus/baumannii complex.

Below this level:
**Acinetobacter baumannii 1656-2** : Acinetobacter baumannii
**Acinetobacter baumannii AB0057** : Acinetobacter baumannii
**Acinetobacter baumannii AB307-0294** : Acinetobacter baumannii
**Acinetobacter baumannii ACICU** : Acinetobacter baumannii
**Acinetobacter baumannii ATCC 17978** : Acinetobacter baumannii
**Acinetobacter baumannii AYE** : Acinetobacter baumannii
**Acinetobacter baumannii BJAB07104** : Acinetobacter baumannii
**Acinetobacter baumannii BJAB0715** : Acinetobacter baumannii
**Acinetobacter baumannii BJAB0868** : Acinetobacter baumannii
**Acinetobacter baumannii D1279779** : Acinetobacter baumannii
**Acinetobacter baumannii MDR-TJ** : Acinetobacter baumannii
**Acinetobacter baumannii MDR-ZJ06** : Acinetobacter baumannii
**Acinetobacter baumannii TCDC-AB0715** : Acinetobacter baumannii
**Acinetobacter baumannii TYTH-1** : Acinetobacter baumannii
**Acinetobacter baumannii ZW85-1** : Acinetobacter baumannii
**Acinetobacter calcoaceticus PHEA-2** : Acinetobacter calcoaceticus

Lineage

Pseudomonadales
Moraxellaceae
Acinetobacter
Acinetobacter calcoaceticus/baumannii complex
Acinetobacter baumannii
Acinetobacter calcoaceticus

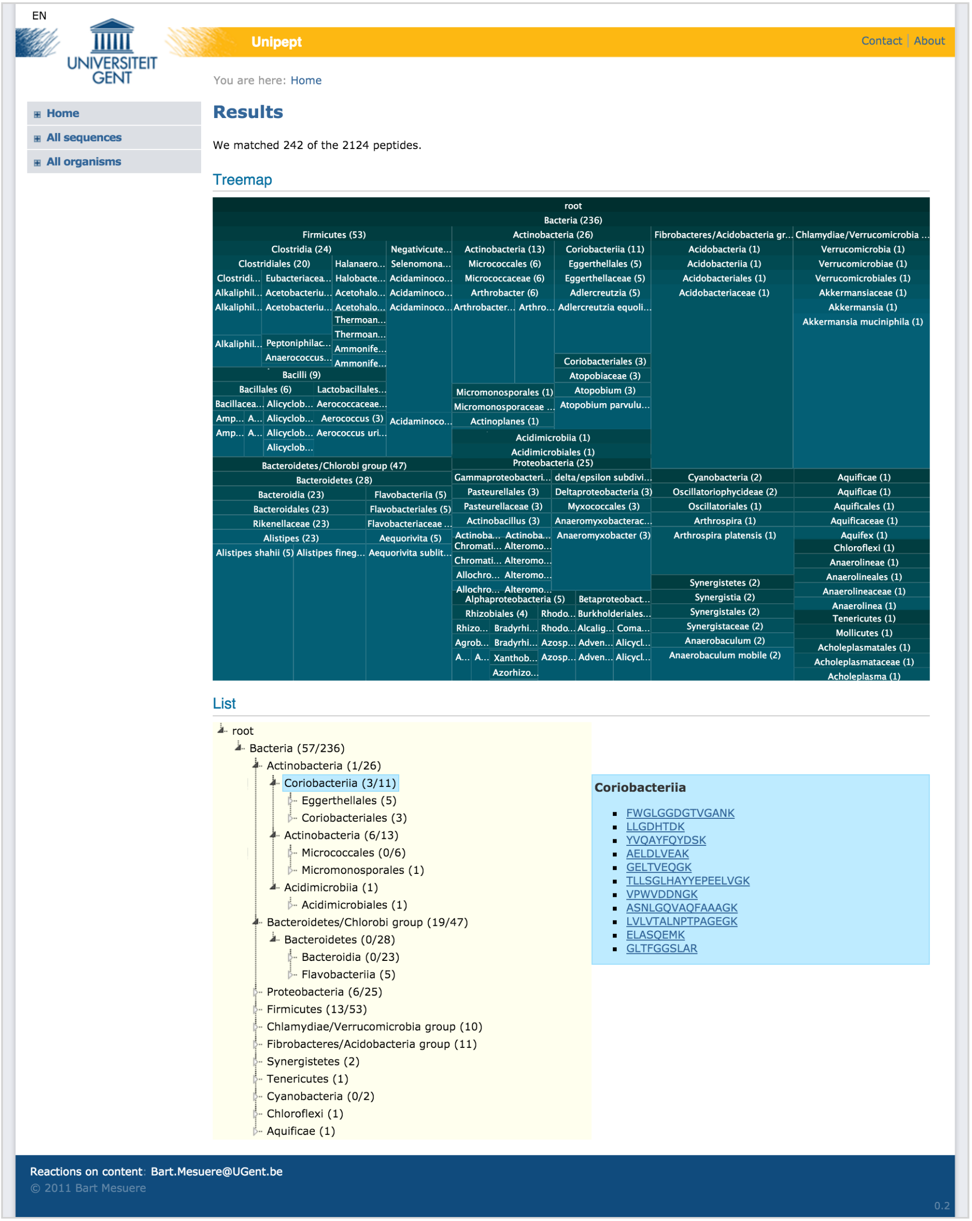
External links

- Blast this peptide

Reactions on content: Bart.Mesuere@UGent.be
© 2011 Bart Mesuere
0.2

**Figure 5.5** Result page of the single peptide analysis in Unipept version 0.2 for peptide EVAEAAQEK.

The next main feature of Unipept became the multi-peptide analysis (Figure 5.6). This feature allowed a user to submit a list of tryptic peptides from a metaproteomics experiment instead of just a single peptide. Listing all occurrences for each tryptic peptide would produce a long and cluttered list of information. Instead, only the LCA of each of the submitted peptides was used. These results were aggregated per taxonomic node into some kind of hierarchical frequency table. Clicking on a node in this table revealed the peptides associated with that taxonomic node. The same information was displayed as in interactive treemap using the JavaScript InfoVis Toolkit (http://philogb.GitHub.io/jit/), a JavaScript visualization framework.



You are here: Home

**Results**

We matched 242 of the 2124 peptides.

Treemap

List

- root
  - Bacteria (57/236)
    - Actinobacteria (1/26)
      - Coriobacteriia (3/11)
        - Eggerthellales (5)
        - Coriobacteriales (3)
      - Actinobacteria (6/13)
        - Micrococcales (0/6)
        - Micromonosporales (1)
      - Acidimicrobiia (1)
        - Acidimicrobiales (1)
    - Bacteroidetes/Chlorobi group (19/47)
      - Bacteroidetes (0/28)
        - Bacteroidia (0/23)
        - Flavobacteriia (5)
    - Proteobacteria (6/25)
    - Firmicutes (13/53)
    - Chlamydiae/Verrucomicrobia group (10)
    - Fibrobacteres/Acidobacteria group (11)
    - Synergistetes (2)
    - Tenericutes (1)
    - Cyanobacteria (0/2)
    - Chloroflexi (1)
    - Aquificae (1)

**Coriobacteriia**

- FWGLGGDGTVGANK
- LLGDHTDK
- YVQAYFQYDSK
- AELDLVEAK
- GELTVEQGK
- TLLSGLHAYYEPEELVGK
- VPWVDDNGK
- ASNLGQVAQFAAAGK
- LVLVTALNPTPAGEGK
- ELASQEMK
- GLTFGGSLAR

Reactions on content: Bart.Mesuere@UGent.be
© 2011 Bart Mesuere
0.2

**Figure 5.6** Result page of the multi-peptide analysis in Unipept version 0.2. The page shows the result of the analysis of Sample 7 as defined by Verberkmoes et al. (2009).

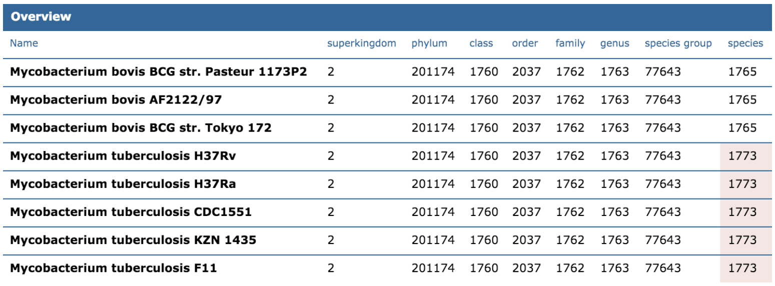
## Unipept version 0.3

Released on July 18, 2011

For Unipept version 0.3, we changed our protein data source from the RefSeq genomes to the UniProt Knowledgebase (Wu et al., 2006). UniProtKB consists of two parts, Swiss-Prot and TrEMBL, containing millions of protein entries, including proteins from complete and reference proteomes. Our Java pipeline iterates over the xml version of UniProt to extract the protein entries and additional metadata such as organism information and various cross references. These additional data are stored in an updated database schema. Switching from RefSeq to UniProt not only yielded more protein data, but also richer annotations.

We implemented a SAX parser because of its streaming property and the huge size of our data file.

From a user's point of view, there were two small additions in Unipept 0.3. The list of organisms in which the tryptic peptides occurs (Figure 5.5) was replaced by a table. This table not only includes the organism name of the matched UniProt entry, but also the complete lineage of that organism (Figure 5.7). The second addition was that of a list of all peptides that could not be matched by Unipept on the multi-peptide analysis result page. These peptides are accompanied by a link to immediately start a BLASTp search for them.


**Overview**

| Name | superkingdom | phylum | class | order | family | genus | species group | species |
|---|---|---|---|---|---|---|---|---|
| **Mycobacterium bovis BCG str. Pasteur 1173P2** | 2 | 201174 | 1760 | 2037 | 1762 | 1763 | 77643 | 1765 |
| **Mycobacterium bovis AF2122/97** | 2 | 201174 | 1760 | 2037 | 1762 | 1763 | 77643 | 1765 |
| **Mycobacterium bovis BCG str. Tokyo 172** | 2 | 201174 | 1760 | 2037 | 1762 | 1763 | 77643 | 1765 |
| **Mycobacterium tuberculosis H37Rv** | 2 | 201174 | 1760 | 2037 | 1762 | 1763 | 77643 | 1773 |
| **Mycobacterium tuberculosis H37Ra** | 2 | 201174 | 1760 | 2037 | 1762 | 1763 | 77643 | 1773 |
| **Mycobacterium tuberculosis CDC1551** | 2 | 201174 | 1760 | 2037 | 1762 | 1763 | 77643 | 1773 |
| **Mycobacterium tuberculosis KZN 1435** | 2 | 201174 | 1760 | 2037 | 1762 | 1763 | 77643 | 1773 |
| **Mycobacterium tuberculosis F11** | 2 | 201174 | 1760 | 2037 | 1762 | 1763 | 77643 | 1773 |

**Figure 5.7** The lineage table that was added in the single peptide analysis result page in Unipept version 0.3

## Unipept version 0.4

Released on May 3, 2012.

Until Unipept 0.4, we always replaced all occurrences of isoleucine (I) by leucine (L) during initial processing. While this makes sense for proteomics data, it makes it im-

Isoleucine and leucine are isobaric amino acids and therefore hard to tell apart in MS experiments.

possible to optionally discern between them in a search because that data simply is not present in the database. One could of course store the original sequence in the database and make the substitution at runtime, but this would have a, potentially severe, negative performance impact. When searching for a peptide with $x$ Is or Ls, we would have to combine the result of searching for all possible I/L-combinations, meaning $2^x$ sequences. Instead, we chose to both store the original sequence and the "equated" version. This way, we can offer the user the choice if he wants to differentiate between I and L or not. The only disadvantage is the doubled storage space that is needed to store the sequence and its index in the database.

A second database change in version 0.4 was the caching of the LCA calculation. Individual LCA calculation is quite fast due to the table-based lineage approach, but when performing a multi-peptide search, calculating the LCA for each of the potentially thousands of input peptides can take some time. Since the LCA doesn't change if the rest of the data isn't changed, it makes sense to store the result after it's been calculated the first time. Because we now give the user the choice of equating isoleucine and leucine and because this influences the outcome of the LCA, a value for both settings must be stored separately. Note that this is the first time that the rails application writes something to the database. Until now, only read access was needed.

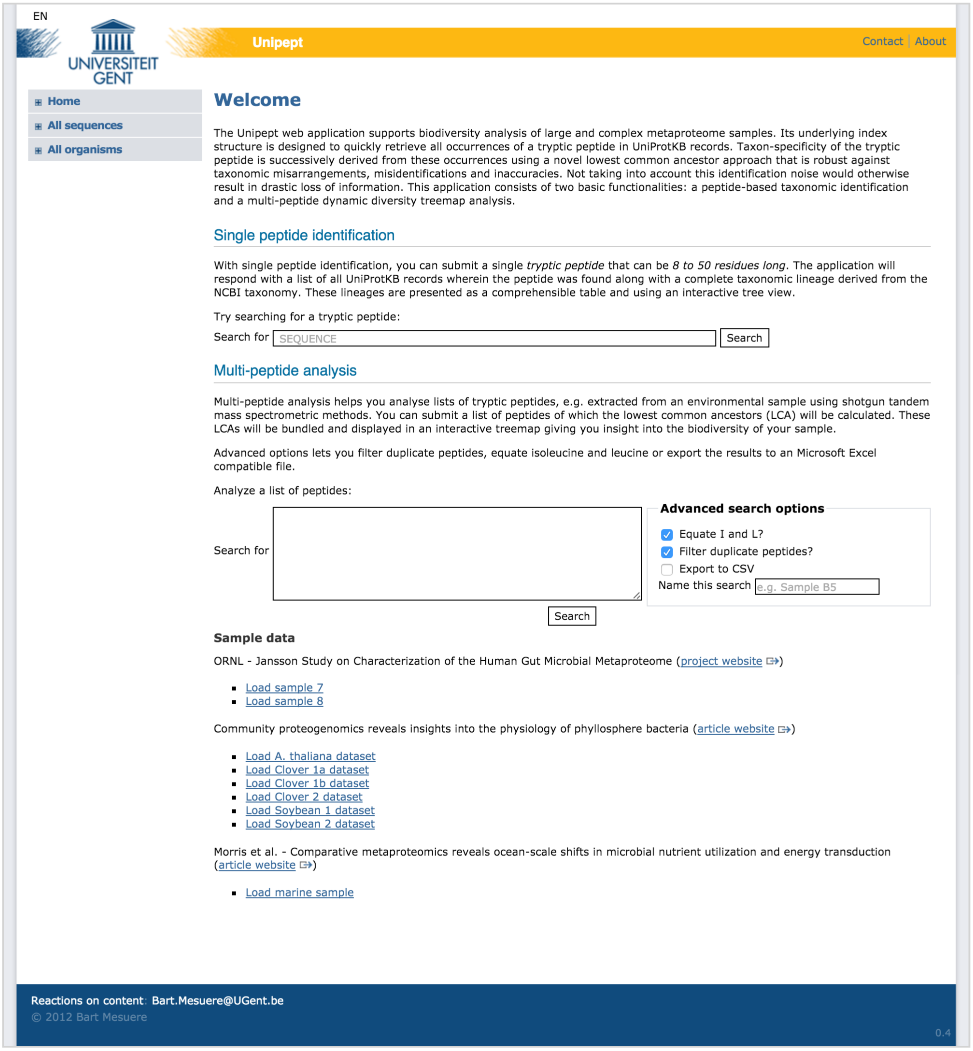
Later versions of Unipept pre-compute the LCA for each peptide as part of the parsing step.

Another change to the database in Unipept 0.4 was the need to store the validity of taxa. This was done by adding an extra column to the taxa table containing a binary value. While suitable for checking the validity of a single taxon, this solution was not sufficient for an efficient LCA calculation which only uses data from the lineage table. Therefore, the taxon validity data had to be included in the lin-

The reasons for and effects of the taxonomy cleanup are discussed in Section 2.1.2.3.

eage table in a way that invalid taxa could easily be ignored during LCA calculation. Our solution was to multiply the invalid taxon id's by -1 before storing them in the lineage table. Although this solution looks like a kludge, it has many benefits: it's easy to ignore these taxa during LCA calculation by ignoring negative taxon id's, it's possible to discern between invalid taxa and missing values, it has a minimal performance impact and it's possible to retrieve the original taxon id by simply taking its absolute value. The invalidation itself is done during the initial parsing.

In the competitive field of bioinformatics, it's important to make the threshold to start using your application as low as possible. Being a web application already eliminates a cumbersome installation step and providing easy access to test data further lowers the barrier. We selected three diverse metaproteomics projects and added one-click access to their data to the Unipept home page (Figure 5.8). Other new features introduced in version 0.4 include the addition of a CSV export option and the display of the full organism names in the single peptide analysis results table instead of the taxon id's.



Welcome

The Unipept web application supports biodiversity analysis of large and complex metaproteome samples. Its underlying index structure is designed to quickly retrieve all occurrences of a tryptic peptide in UniProtKB records. Taxon-specificity of the tryptic peptide is successively derived from these occurrences using a novel lowest common ancestor approach that is robust against taxonomic misarrangements, misidentifications and inaccuracies. Not taking into account this identification noise would otherwise result in drastic loss of information. This application consists of two basic functionalities: a peptide-based taxonomic identification and a multi-peptide dynamic diversity treemap analysis.

Single peptide identification

With single peptide identification, you can submit a single *tryptic peptide* that can be *8 to 50 residues long*. The application will respond with a list of all UniProtKB records wherein the peptide was found along with a complete taxonomic lineage derived from the NCBI taxonomy. These lineages are presented as a comprehensible table and using an interactive tree view.

Try searching for a tryptic peptide:

Search for SEQUENCE Search

Multi-peptide analysis

Multi-peptide analysis helps you analyse lists of tryptic peptides, e.g. extracted from an environmental sample using shotgun tandem mass spectrometric methods. You can submit a list of peptides of which the lowest common ancestors (LCA) will be calculated. These LCAs will be bundled and displayed in an interactive treemap giving you insight into the biodiversity of your sample.

Advanced options lets you filter duplicate peptides, equate isoleucine and leucine or export the results to an Microsoft Excel compatible file.

Analyze a list of peptides:

Search for

Advanced search options

- [✓] Equate I and L?
- [✓] Filter duplicate peptides?
- [ ] Export to CSV

Name this search e.g. Sample B5

Search

Sample data

ORNL - Jansson Study on Characterization of the Human Gut Microbial Metaproteome (project website)

- Load sample 7
- Load sample 8

Community proteogenomics reveals insights into the physiology of phyllosphere bacteria (article website)

- Load A. thaliana dataset
- Load Clover 1a dataset
- Load Clover 1b dataset
- Load Clover 2 dataset
- Load Soybean 1 dataset
- Load Soybean 2 dataset

Morris et al. - Comparative metaproteomics reveals ocean-scale shifts in microbial nutrient utilization and energy transduction (article website)

- Load marine sample

Reactions on content: Bart.Mesuere@UGent.be
© 2012 Bart Mesuere
0.4

**Figure 5.8** Home page of Unipept 0.4 showing additional documentation and the availability sample datasets.

## 5.2.2 Unipept version 1.0

Released on May 29, 2012.

After the release of Unipept 0.4, everything was prepared to release the first production-ready version. This means that the main focus was on bug fixes, layout tweaks and more documentation. The only new feature was the inclusion of the sunburst graph, a new visualization that had been in development for a while but wasn't ready yet for release with Unipept 0.4.

GitHub Enterprise is a self-hosted version of the public GitHub.com website.

During the development of version 1.0, Ghent University deployed GitHub Enterprise. This allowed all students and researchers to create an unlimited number of (private) repositories to manage their programming projects free of charge. Since the Unipept team was very much in favor of offering GitHub Enterprise as a university-wide repository hosting service, we immediately switched over as a pilot user. This switch resulted in a more professional development approach and the adoption of several best-practices.

### The flow branching model

The most profound change was the adoption of the flow branching model (Figure 5.9) instead of working on a single branch. Flow uses two core branches: `develop` and `master`. All new development work should happen on `develop`, while `master` always reflects a production-ready state. Once enough work is done on `develop` to justify a new release, the branch is merged into `master`, a version tag is created and the new version can be deployed on the production servers.

Next to these two core branches, two types of temporary, supporting branches are used: feature branches and hotfix branches. As the name implies, feature branches are used to experiment and develop new features without interfering with other development. They always start by branching of `develop` and are merged back into `develop` when the feature is finished. A hotfix branch is used to fix a critical bug in the current production version. When such bug is discovered, a new branch is started from `master`, the bug gets fixed and the hotfix branch gets merged into both `master` and `develop` resulting in a new release.

The flow branching model also suggests using release branches before creating a new release and merging changes into `master`. This release branch can be used for final testing, documentation generation, last-minute bug fixes and other release-oriented tasks without interrupting continuing work on `develop`. Because Unipept development is mostly a one-man operation, this type of branch was not really useful for us.

**Issues, pull requests & releases**

The flow guidelines note that feature branches typically only exist in developer repos and are only pushed to the main repository when the feature is finished. This is not something we agree with, on the contrary, we encourage pushing feature branches as early as possible to be able to create a pull request. A pull request was originally meant to be a way to submit modifications to an open source project. On GitHub, any branch can be used to create a pull request. This creates some sort of timeline view of the branch, containing all commits and comments chronologically. In the Unipept repository, we use these discussion views to document the development of the feature by commenting with motivations for design choices, benchmark results, screenshots, task lists, etc. Other benefits of using pull requests over simple branches are that they can be labeled, are searchable and can be linked to. Branches also eventually get deleted, pull requests are never deleted, only closed or merged.

The pull requests and changelog are what made this reconstruction of the Unipept history possible.

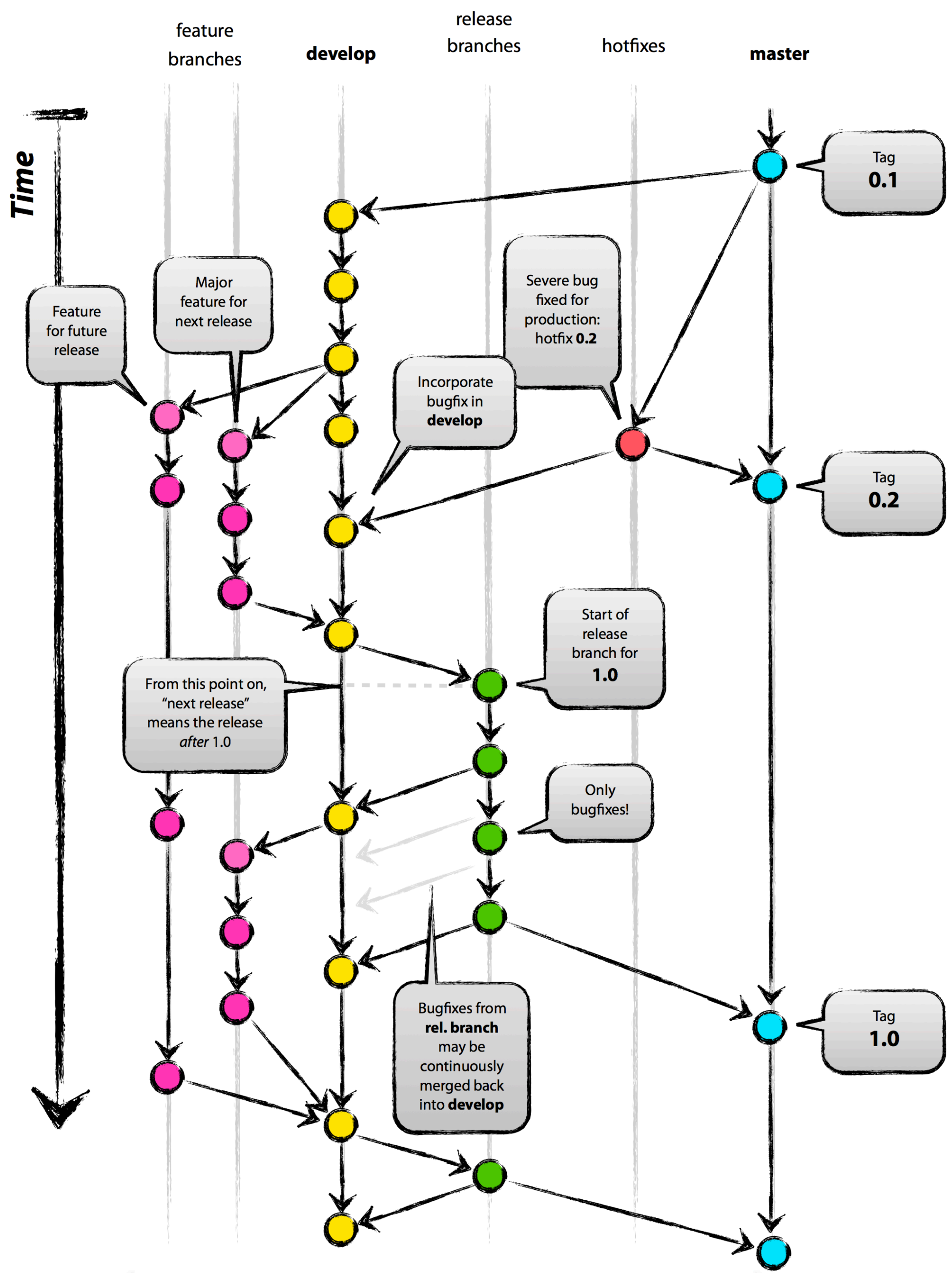

feature branches
develop
release branches
hotfixes
master
Time
Tag 0.1
Feature for future release
Major feature for next release
Severe bug fixed for production: hotfix 0.2
Incorporate bugfix in develop
Tag 0.2
Start of release branch for 1.0
From this point on, "next release" means the release after 1.0
Only bugfixes!
Bugfixes from rel. branch may be continuously merged back into develop
Tag 1.0

**Figure 5.9** Schematic overview of the flow branching model. Image created by Vincent Driessen

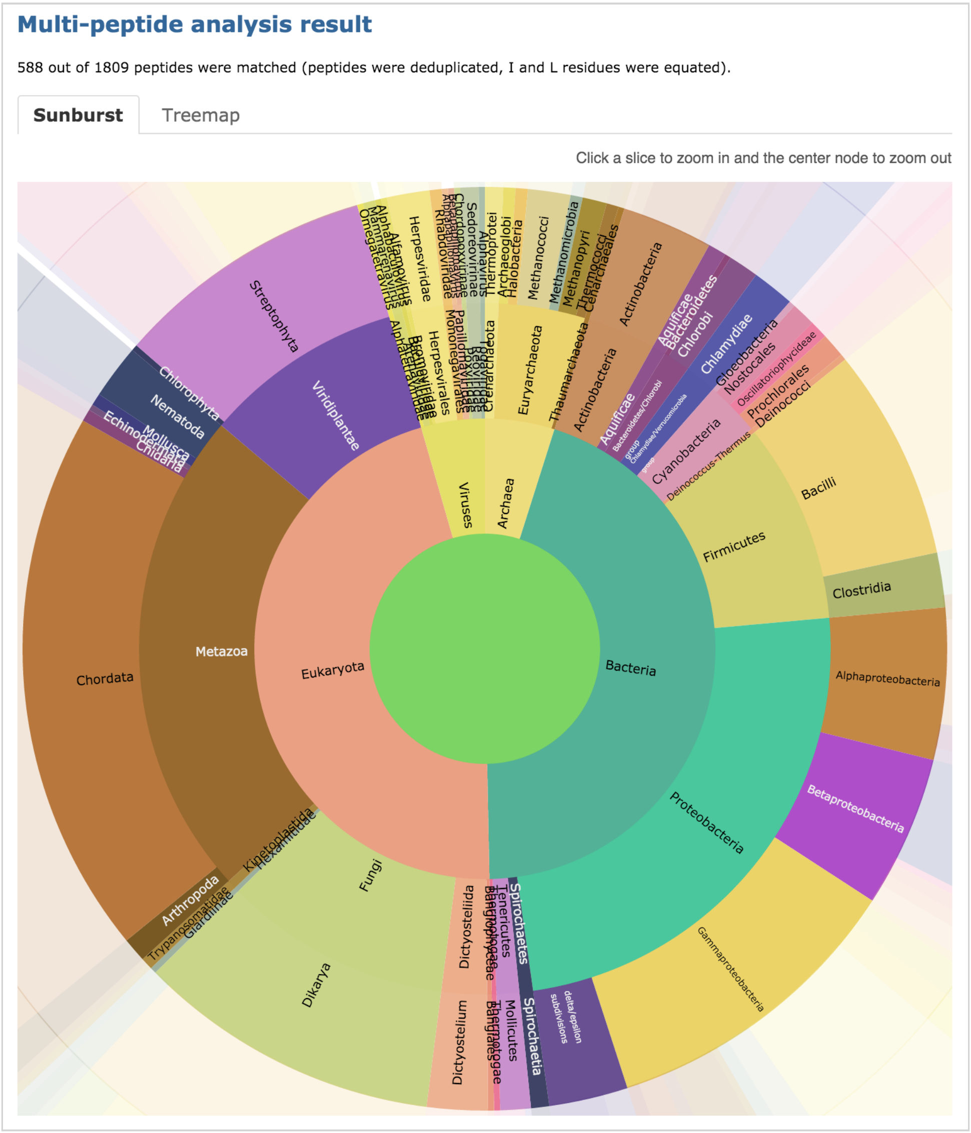
The Unipept repository used over 500 issues for around 3000 commits.

Another GitHub feature that helped the management of the project is the extensive use of issues. Issues is the central bug tracker of GitHub that can be used for bug reports, but also to keep track of ideas for features and other tasks. Each issue gets assigned an incremental numeric identifier after creation. This id can then be used to reference the respective issue in commit messages and throughout the GitHub website. Just like pull requests, issues can also be labeled and assigned to milestones. We always create milestones for the next several Unipept versions and by assigning issues to them, we create a coarse roadmap. This way, the list of all open issues for the next milestone release can serve as a to-do list. When all issues for a milestone are closed (or moved to the next milestone), a new Unipept version is released. The list of closed issues for that milestone can then be used to create a changelog. From version 1.0, this changelog is published on both GitHub and the Unipept website.

After a new version is released, it must also be deployed on the Unipept production servers. To automate this process and minimize downtime, we use a tool called Capistrano (http://capistranorb.com/). Capistrano uses our GitHub repository to automatically download the latest version of the code to the server, it updates the dependencies and takes care of asset generation. By default, our Capistrano configuration deploys the `master` branch on production machines and the `develop` branch on our test servers.

### D3.js

The only new feature in Unipept 1.0 is the inclusion of a new type of visualization: the sunburst graph (Figure 5.10). The sunburst is an interactive multi-level pie chart where the center represents the root node with several concentric rings around it. These rings are divided into slices representing the nodes in the taxonomy and are aligned according to their hierarchical relation. The size of the slices corresponds to the number of peptides having an LCA equal to that taxonomic node or any of its children.



Multi-peptide analysis result

588 out of 1809 peptides were matched (peptides were deduplicated, I and L residues were equated).

**Figure 5.10** Part of the results page of the multi-peptide analysis in Unipept 1.0 showing the newly added sunburst diagram.

D3 is short for Data-Driven Documents.

The sunburst graph was implemented using D3.js (Bostock, Ogievetsky, and Heer, 2011), a JavaScript framework to create dynamic, interactive data visualizations in web browsers. D3 doesn't come with pre-built charts and visualizations, but requires you to build them yourself. To help with this, it includes several helper methods such as functions to draw axes, map data from a domain to a different range, or calculate complex layouts. The main drawback of D3 is its steep learning curve, but once you've got it figured out, you have ultimate control over the design, animations, and interactions of your visualization.

DOM stands for Document Object Model, a convention for working with elements in HTML and XML documents.

The main idea of D3 is to bind data to the DOM of a web page and then use that data to apply data-driven transformations. For example, you could represent a dataset as a set of circles, where a categorical property of each data point controls the color of the circle, a nominal property controls its radius and another its position. When new data becomes available, the bound data can be updated and the corresponding transformations can adapt the visualization using an animated transition. While D3 is mostly used in combination with SVG graphics, it works with any HTML element.

The sunburst data format is similar to the treemap format, but is generated separately. Both formats are unified in Unipept 2.5.

For the sunburst graph, the input is a hierarchical JSON object generated by the Ruby on Rails middleware. Each `node` object contains information on a single taxonomic node, such as name, rank, and number of associated peptides, and a list of children. These children are its taxonomic descendants and are also `node` objects. This hierarchical data structure is processed by the D3 partition layout algorithm to convert the hierarchical data and the number of peptides to a set of *x*- and *y*-coordinates. These coordinates are then used as angles and radii to draw the diagram using SVG elements.

## 5.2.3 Unipept version 1.1 – 1.5

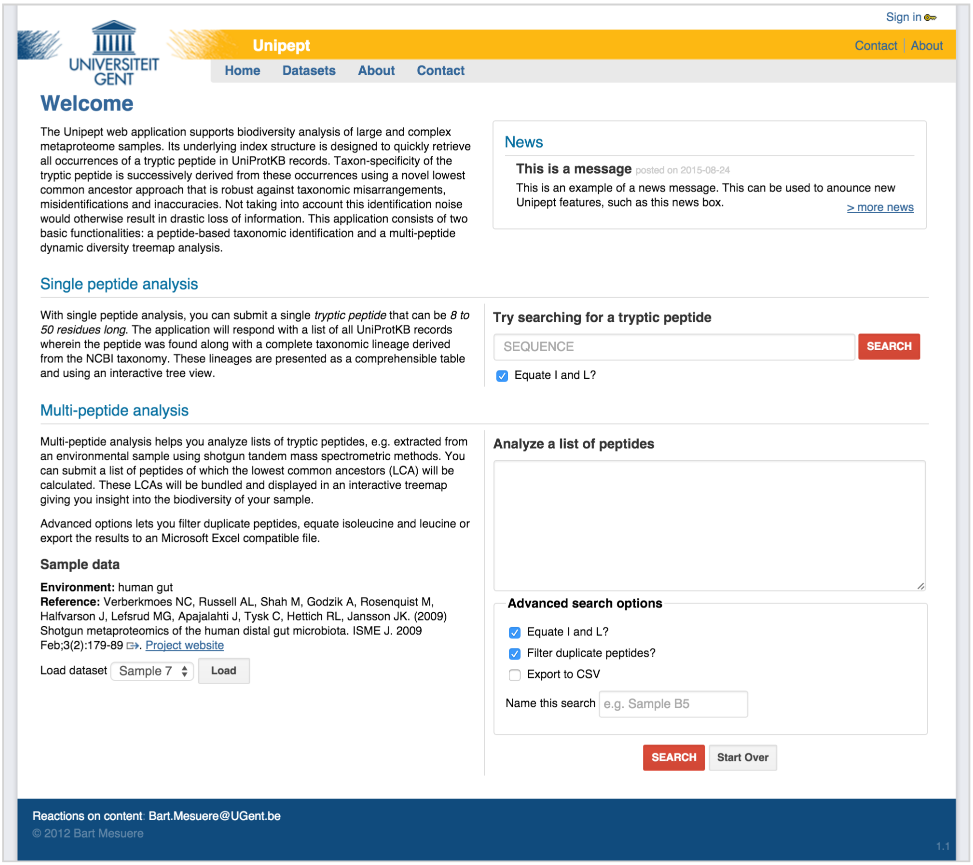
Unipept 1.0 can be seen as a minimum viable product; it contained all basic features that are needed to be a useful research tool, but not much more. This is also reflected by the fact that the first Unipept article (Mesuere et al., 2012) was based on that version. Its release was followed by a number of smaller updates over the next 6 months. With the exception of support for missed cleavages, no major new features were added. Instead, the focus was on improving the existing application by paying more attention to design, usability, and performance.

### Unipept version 1.1

Released on June 18, 2012.

The most visible change of Unipept 1.1 was the reworked design (Figure 5.11). The navigation on the left was dropped in favor of a smaller navigation bar in the header of each page. Since a fixed-width page design was used, this left more space for the actual content, something that was especially beneficial for the visualizations. Another change was the addition of news items to inform users of new features and database updates. A news box was added to the home page in which the latest news item was shown.

From a technical point of view, the biggest change was the update from rails 3.0 to rails 3.2. This update introduced the rails asset pipeline in Unipept. The asset pipeline is a framework to manage CSS and JavaScript resources in a web application. Concretely, this means that when deploying your application to a production environment, your assets are “compiled”. During this compilation step, all CSS files of your application are bundled into a single file whilst removing unnecessary white-space. Similarly, all JavaScript files are bundled into one file which is then minified. As the name implies, minification tries to make the input file as small as possible by removing white-space and comments, and by renaming variables and functions to shorter names. The advantage of using the asset pipeline is that as a developer, you can use as many files as you want to organize your code without suffering from the related impact on performance.



Welcome

The Unipept web application supports biodiversity analysis of large and complex metaproteome samples. Its underlying index structure is designed to quickly retrieve all occurrences of a tryptic peptide in UniProtKB records. Taxon-specificity of the tryptic peptide is successively derived from these occurrences using a novel lowest common ancestor approach that is robust against taxonomic misarrangements, misidentifications and inaccuracies. Not taking into account this identification noise would otherwise result in drastic loss of information. This application consists of two basic functionalities: a peptide-based taxonomic identification and a multi-peptide dynamic diversity treemap analysis.

News

**This is a message** posted on 2015-08-24

This is an example of a news message. This can be used to anounce new Unipept features, such as this news box.

> more news

Single peptide analysis

With single peptide analysis, you can submit a single *tryptic peptide* that can be *8 to 50 residues long*. The application will respond with a list of all UniProtKB records wherein the peptide was found along with a complete taxonomic lineage derived from the NCBI taxonomy. These lineages are presented as a comprehensible table and using an interactive tree view.

**Try searching for a tryptic peptide**

SEQUENCE SEARCH

✓ Equate I and L?

Multi-peptide analysis

Multi-peptide analysis helps you analyze lists of tryptic peptides, e.g. extracted from an environmental sample using shotgun tandem mass spectrometric methods. You can submit a list of peptides of which the lowest common ancestors (LCA) will be calculated. These LCAs will be bundled and displayed in an interactive treemap giving you insight into the biodiversity of your sample.

Advanced options lets you filter duplicate peptides, equate isoleucine and leucine or export the results to an Microsoft Excel compatible file.

**Sample data**

**Environment:** human gut
**Reference:** Verberkmoes NC, Russell AL, Shah M, Godzik A, Rosenquist M, Halfvarson J, Lefsrud MG, Apajalahti J, Tysk C, Hettich RL, Jansson JK. (2009) Shotgun metaproteomics of the human distal gut microbiota. ISME J. 2009 Feb;3(2):179-89. Project website

Load dataset Sample 7 Load

**Analyze a list of peptides**

**Advanced search options**

✓ Equate I and L?
✓ Filter duplicate peptides?
☐ Export to CSV

Name this search e.g. Sample B5

SEARCH Start Over

Reactions on content: Bart.Mesuere@UGent.be
© 2012 Bart Mesuere
1.1

**Figure 5.11** The home page of Unipept 1.1.

### Unipept version 1.2

Released on July 9, 2012.

Unipept 1.2 contained no visible changes and solely focused on performance improvements. As mentioned in a previous side note, both the treemap and sunburst visualizations use a JSON object as their data source. These two similar JSON objects are first created as a single Ruby root node object that contains all data for the two visualizations. Since the desired output format is slightly different for both, they must be generated separately which involves a few back and forth conversion to the JSON format. These conversion steps accounted for a majority of the 2500 ms page load time. Swapping out the default Ruby JSON parser for OJ, a JSON parser optimized for speed, reduced loading time to only 500 ms.

The initial root object is essentially deep copied by serialization.

A second set of performance improvements consisted of optimizing all queries for performance and applying eager loading where possible. Eager loading is a Ruby on Rails mechanism where associated records are loaded in as few queries as possible. For example, retrieving all UniProt entries in which a given peptide occurs can be done in a single query. If we afterwards want to fetch information on the associated taxonomy records, we need one query per UniProt entry. If we know in advance that we will need the taxonomy data, we can use eager loading to fetch that data while doing the initial query using only a single extra query.

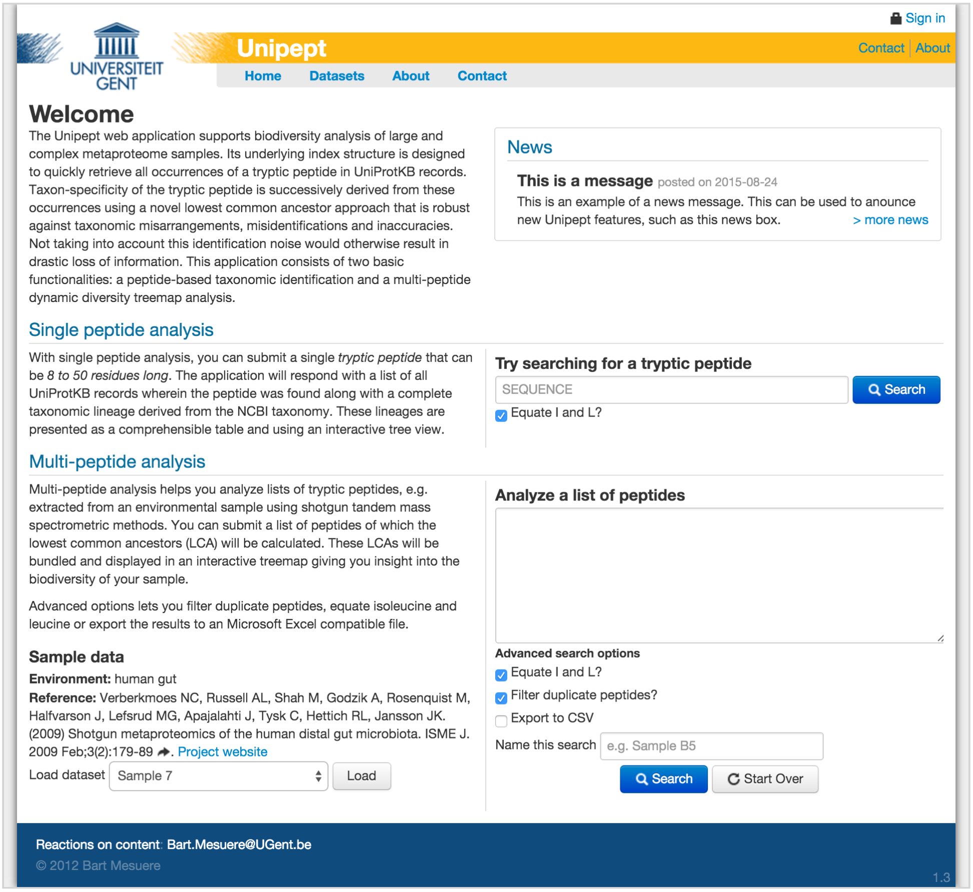
This is called the N+1 query problem: doing one initial query + one for each of the N associations.

### Unipept version 1.3

Released on August 27, 2012.

Unipept 1.3 further tweaked the user interface of the website by using Twitter Bootstrap (Figure 5.12). Bootstrap is a popular, open source front-end framework to create web applications. It contains HTML and CSS templates, for example for typography, icons, forms, etc., and a number of JavaScript powered components such as modal dialogs and tooltips. The main benefit of using Bootstrap is that it speeds up front-end development because of its ready-made components that don't require any additional styling to create an ok-looking, but somewhat generic, web page.

The project is no longer associated with Twitter and is now called Bootstrap.



Welcome

The Unipept web application supports biodiversity analysis of large and complex metaproteome samples. Its underlying index structure is designed to quickly retrieve all occurrences of a tryptic peptide in UniProtKB records. Taxon-specificity of the tryptic peptide is successively derived from these occurrences using a novel lowest common ancestor approach that is robust against taxonomic misarrangements, misidentifications and inaccuracies. Not taking into account this identification noise would otherwise result in drastic loss of information. This application consists of two basic functionalities: a peptide-based taxonomic identification and a multi-peptide dynamic diversity treemap analysis.

News

**This is a message** posted on 2015-08-24
This is an example of a news message. This can be used to anounce new Unipept features, such as this news box. > more news

Single peptide analysis

With single peptide analysis, you can submit a single *tryptic peptide* that can be *8 to 50 residues long*. The application will respond with a list of all UniProtKB records wherein the peptide was found along with a complete taxonomic lineage derived from the NCBI taxonomy. These lineages are presented as a comprehensible table and using an interactive tree view.

**Try searching for a tryptic peptide**
SEQUENCE [Search]
[✓] Equate I and L?

Multi-peptide analysis

Multi-peptide analysis helps you analyze lists of tryptic peptides, e.g. extracted from an environmental sample using shotgun tandem mass spectrometric methods. You can submit a list of peptides of which the lowest common ancestors (LCA) will be calculated. These LCAs will be bundled and displayed in an interactive treemap giving you insight into the biodiversity of your sample.

Advanced options lets you filter duplicate peptides, equate isoleucine and leucine or export the results to an Microsoft Excel compatible file.

**Sample data**
**Environment:** human gut
**Reference:** Verberkmoes NC, Russell AL, Shah M, Godzik A, Rosenquist M, Halfvarson J, Lefsrud MG, Apajalahti J, Tysk C, Hettich RL, Jansson JK. (2009) Shotgun metaproteomics of the human distal gut microbiota. ISME J. 2009 Feb;3(2):179-89. Project website
Load dataset: Sample 7 [Load]

**Analyze a list of peptides**

**Advanced search options**
[✓] Equate I and L?
[✓] Filter duplicate peptides?
[ ] Export to CSV
Name this search: e.g. Sample B5
[Search] [Start Over]

Reactions on content: Bart.Mesuere@UGent.be
© 2012 Bart Mesuere
1.3

**Figure 5.12** The home page of Unipept 1.3, the first release to use Bootstrap.

A second focus area of Unipept 1.3 was to improve the usability of the visualizations by implementing a fullscreen mode and by allowing users to export the visualizations as an image. The Fullscreen API in browsers is a *living standard* which means that it is still in active development and not a finalized standard. As a result, it's impossible for browsers to adhere to the standard since there is none. All major browsers do have their own implementation of a fullscreen API, but unfortunately they are not mutually compatible and incompatible with the latest draft of the spec. To reliably implement fullscreen support, we had to write our own mini API containing the necessary functions as a compatibility layer. This compatibility layer then called different internal functions depending on which browser (version) was used. The downside of this approach is that to maintain compatibility, we had to keep an eye on browsers changing their implementation.

At the time, Unipept contained two types of visualizations: the treeview and treemap are based on the canvas element, while the sunburst uses the svg element. Exporting them as images required a different approach for each. The solution for the canvas visualizations was very simple: the canvas API contains a `toDataURL`-method that returns the contents of the canvas as an image. The svg API has no similar alternative and required a server-side solution. When the user clicks the *save image* button, the content of the svg element gets sent to the server. There, ImageMagick is then called to rasterize the vector graphic and the resulting image gets Base64 encoded and sent back to the client as a data-url. Although these two procedures are very different in nature, they happen invisible to the user.

ImageMagick is a set of command line utilities for manipulating images.

### Unipept version 1.4

Released on October 11, 2012.

Unipept 1.4 was released as a response to some of the review remarks of the initial Unipept article (Mesuere et al., 2012). Up until now, our processing pipeline only stored peptides with a length between 8 and 50 amino acids. At the request of one of the reviewers, this was expanded to a length between 5 and 50. Other than an increased database size, this had no implications on the application.

A second request was the addition of missed cleavage handling. When trypsin is added to a protein sample, it hasn't got a 100% success rate. Sometimes it misses a cleavage site, resulting in peptides consisting of two or more tryptic parts. Since Unipept only keeps an index of tryptic peptides, these composite peptides cannot be found using Unipept. A pragmatic solution to this is to do an in-silico tryptic digest after the peptide has been submitted to Unipept and to search for the two or more separate parts. This technique has no negative performance impact and will never produce incorrect results since the matched peptides were genuinely present in the sample. It is however not the most accurate solution, because we don't take into account the fact that the separate parts of the peptide are adjacent in the same protein.

The easiest solution to this would be to not only store tryptic peptides, but also peptides with one or more missed cleavages. The big drawback is that the database size would grow exponentially, depending on how many missed cleavages we wanted to support. Since this is not desirable, we came up with a better solution which we call advanced missed cleavage handling. We still do an in silico digest of the submitted peptide, but instead of returning the various parts as separate peptides, we recombine the results. Since the separate parts occur in the same peptide, we only want UniProt entries where each of the parts occur. The first step in the recombination is thus to take the intersection of the sets of matched UniProt entries for the separate peptide parts. We now have a reduced set of UniProt entries that potentially contain our composite peptide. The second step is to account for the locality since the separate parts occur adjacently. We do this by performing a full text search on the composite peptide in each of the protein sequences of the reduced set of UniProt entries and only retaining those entries with a match. The LCA for the composite peptide is then calculated as normally from the remaining UniProt entries. The biggest downside of this approach is that it takes some time to do the extra calculations, which is why the advanced missed cleavage handling is disabled by default. The big advantage is that it works for any number of missed cleavages.

### Unipept version 1.5

Released on March 15, 2013.

PRIDE stands for PRoteomics IDEntifications.

The main feature of Unipept 1.5 was support for loading datasets from PRIDE, a public data repository for proteomics data. When a user enters the id of a PRIDE experiment into the Unipept website, the Unipept server fetches the corresponding dataset from PRIDE using BioMart and then sends the peptides back to the user. Data access through BioMart proved very cumbersome and unnecessary difficult. To request data using the BioMart API, one must construct an xml file containing the desired input and output using a special tool. The generated file must then be URL-encoded and added to the request-URL as a parameter. The PRIDE BioMart was retired in November 2014 and replaced by the much more user-friendly PRIDE web services. Support for these web services was added in Unipept 2.4.2.

## 5.2.4 Unipept version 2.0

Released on October 29, 2013.

Where Unipept 1.0 can be seen as a minimal version of Unipept, Unipept 1.5 is a more complete version. Now that the dust has settled and the code base has stabilized, it was a good time to review some of the implementation choices and refactor the code. From Unipept 2.0 onwards we aimed for bigger releases, every four to six months, containing at least one big new feature. The big feature of Unipept 2.0 was the addition of the unique peptide finder.

### Spring cleaning

Ruby on Rails is a web application framework using the Model-View-Controller (MVC) principle. MVC is a software pattern for implementing user interfaces that is traditionally used in desktop applications. In this pattern, the application consists of three kinds of components with well-defined interactions between them. A model is used to store data, a controller manipulates the data in the model and a view displays the data of the models. In web applications, the MVC pattern is implemented by letting (a method of) the controller handle incoming requests. The controller can then use models to process the request and prepare data for use in the view. This view consists of a template that gets rendered as a response, for example an HTML page. Models are generally supported by an object-relational mapping (ORM), a system to connect the objects of an application to tables in a relational database. This ORM allows easy access to the properties and relationships of objects in an application that are stored in a database, without writing SQL statements.

In previous versions of Unipept, models were underused and most of the logic was present in the controllers. Additionally, the logic contained lots of SQL queries. To improve the reuse of code, much of the business logic was shifted from the controllers to the models in Unipept 2.0. The numerous queries were eliminated by making use of the ORM features of Ruby on Rails. These improvements reduce the number of code adjustments that are needed when, for example, the database schema changes.

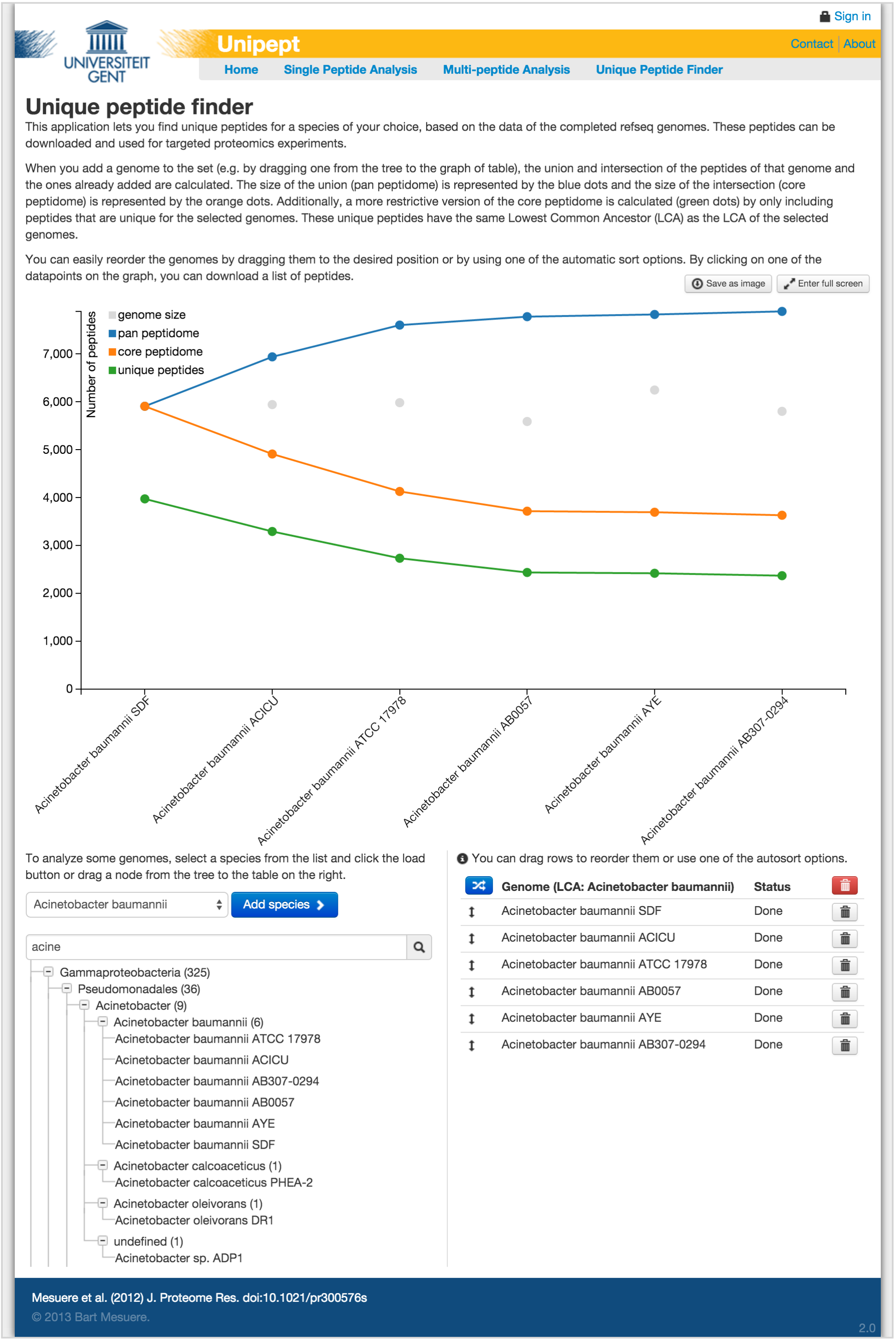
A principle that is know as fat models, thin controllers.

A database change in Unipept 2.0 is the added support for the storage of additional UniProt cross references. Until then, only NCBI and EMBL cross references were stored. These were expanded with EC numbers (Bairoch et al., 2005) and Gene Ontology annotations (Ashburner et al., 2000) with the purpose of adding a functional analysis component to Unipept. A second change was the addition of information about the completed RefSeq genomes. This was done by adding a single table containing the mapping from BioProject identifiers to the existing UniProt cross references.

### Unique peptide finder

See Section 4.1 for more information on the unique proteome.

The unique peptide finder (Figure 5.13) introduced in Unipept 2.0 is the biggest new feature in Unipept history. The associated pull request consists of almost 300 commits spanning a ten-month period. The initial idea for this feature was to replicate something like the microbial core and pan-genome (Medini et al., 2005), but based on the peptide contents of the genomes. Computationally, this comes down to computing unions and intersections for a list of sets, each containing 50 000 to 150 000 short strings. Performance-wise, doing these calculations server-side makes sense. The data is locally accessible from the database, set operations in Ruby are relatively performant and only the results need to be transmitted over the network to the user. The downside is that when embedded in an interactive web application, we would have to give up some flexibility because adding a new genome to the analysis would mean that all previous calculations need to be redone. This is because keeping track of the state of every user on the server is not feasible.


Sign in

Unipept

Contact | About

Home   Single Peptide Analysis   Multi-peptide Analysis   Unique Peptide Finder

UNIVERSITEIT GENT

## Unique peptide finder

This application lets you find unique peptides for a species of your choice, based on the data of the completed refseq genomes. These peptides can be downloaded and used for targeted proteomics experiments.

When you add a genome to the set (e.g. by dragging one from the tree to the graph of table), the union and intersection of the peptides of that genome and the ones already added are calculated. The size of the union (pan peptidome) is represented by the blue dots and the size of the intersection (core peptidome) is represented by the orange dots. Additionally, a more restrictive version of the core peptidome is calculated (green dots) by only including peptides that are unique for the selected genomes. These unique peptides have the same Lowest Common Ancestor (LCA) as the LCA of the selected genomes.

You can easily reorder the genomes by dragging them to the desired position or by using one of the automatic sort options. By clicking on one of the datapoints on the graph, you can download a list of peptides.

Save as image
Enter full screen
Number of peptides
genome size
pan peptidome
core peptidome
unique peptides
0
1,000
2,000
3,000
4,000
5,000
6,000
7,000
Acinetobacter baumannii SDF
Acinetobacter baumannii ACICU
Acinetobacter baumannii ATCC 17978
Acinetobacter baumannii AB0057
Acinetobacter baumannii AYE
Acinetobacter baumannii AB307-0294

To analyze some genomes, select a species from the list and click the load button or drag a node from the tree to the table on the right.

- Gammaproteobacteria (325)
  - Pseudomonadales (36)
    - Acinetobacter (9)
      - Acinetobacter baumannii (6)
        - Acinetobacter baumannii ATCC 17978
        - Acinetobacter baumannii ACICU
        - Acinetobacter baumannii AB307-0294
        - Acinetobacter baumannii AB0057
        - Acinetobacter baumannii AYE
        - Acinetobacter baumannii SDF
      - Acinetobacter calcoaceticus (1)
        - Acinetobacter calcoaceticus PHEA-2
      - Acinetobacter oleivorans (1)
        - Acinetobacter oleivorans DR1
      - undefined (1)
        - Acinetobacter sp. ADP1

You can drag rows to reorder them or use one of the autosort options.

| Genome (LCA: Acinetobacter baumannii) | Status |
|---|---|
| Acinetobacter baumannii SDF | Done |
| Acinetobacter baumannii ACICU | Done |
| Acinetobacter baumannii ATCC 17978 | Done |
| Acinetobacter baumannii AB0057 | Done |
| Acinetobacter baumannii AYE | Done |
| Acinetobacter baumannii AB307-0294 | Done |

Mesuere et al. (2012) J. Proteome Res. doi:10.1021/pr300576s

© 2013 Bart Mesuere.

2.0

**Figure 5.13** The unique peptide finder as introduced in Unipept 2.0.

Shifting all calculations client-side would allow for a more interactive application but involved some challenges: the complete contents of all selected genomes must be transferred to the user, JavaScript has no built-in Set data type, and JavaScript is single-threaded meaning that the browser becomes unresponsive during computations. The absence of a real set forced us to think out of the box to efficiently implement the union and intersection operations. We ultimately decided to use arrays instead and to iterate over them to determine the common elements. If both arrays are sorted, this can be done in a single pass in $O(n + m)$ time where $n$ and $m$ are the length of the arrays. This is comparable to the theoretical complexity of the operations if both were sets instead of sorted arrays. The sorting comes at no extra cost because this is how the entries are returned by the database.

Support for sets was added in ES2015, an updated JavaScript standard finalized in June 2015.

This implementation made us realize that the actual peptide contents of each genome is not needed in the browser to calculate the unions and intersections. Something to uniquely identify the peptides is all that is necessary and the primary keys of the `sequences` database table perfectly fit the bill. The switch from using the actual peptides to using their numerical identifiers had several benefits. Because integers take up much less space than strings, the problem of transferring the genomes from the server to the client was mostly solved: the integer representation is only 150 KB in size for a single bacterial genome. This reduced size benefit is also applicable to the memory needed to store the genomes in the user's browser. A third benefit is that comparing integers is more efficient than strings, making the solution even faster.

With the first two challenges solved, the only remaining issue was the single-threadedness of JavaScript. All

JavaScript code on a web page executes in a single thread which means that for example, the page can't react to button clicks while another calculation is going on. The JavaScript engine keeps a queue of all work that needs to be done and only starts working on the next item when the previous work is finished. To achieve the perception of a smooth, jank free application, a JavaScript developer must make sure that the workloads added to the queue are sufficiently small. Rendering a fluent animation requires a constant frame rate of 60 frames per second or one frame per 16 ms. This means that while an animation is playing, you must make sure that all scheduled work can be executed in under 16 ms or the next frame will be late, resulting in jank. Another example is the responsiveness of the application when interacting with user interface elements. User experience research shows that users expect a response from direct manipulation within 100 ms (Miller, 1968). This means that any calculations that are needed to respond to the click of a button must be done within that time budget.

Jank is any perceived stutter while using an application and has a negative impact on the user experience.

Performing all required calculations within the given time budgets would mean setting up an advanced scheduling system to split up the work. Such solution would rapidly become a complex, hard to maintain, and therefore undesirable affair. Modern browsers offer an alternative solution in the form of web workers. Web workers are a mechanism in which JavaScript code can be run in a background thread. Although it might look like it, this is no real multithreading. The code running in the web worker runs entirely separate from the main JavaScript thread and has no access to the DOM or shared memory access. The only way to communicate between the worker thread and the main thread is by using a simple message passing system. In our case, the time-consuming computations can easily be separated from the rest of the application and communication between both parts is minimal which makes web workers an appropriate solution.

### Single-page application

Once all back-end problems were solved, the design of the application itself could be finalized. The unique peptide finder was designed as a single-page application. This means that the web page behaves as a traditional desktop application without reloading the page or using any page navigations. The user interface itself consists of three main parts: a collapsible tree view containing all available genomes, a table containing all genomes that were added to the analysis and the pan-core graph itself. All three components use D3 to generate HTML and SVG elements.

The collapsible tree to add genomes to the application is generated from a JSON object that is part of the initial page load. The nested JSON object is converted to a nested unordered list using D3, and CSS is applied to make it look like a Windows explorer-like collapsible tree. The collapsing behavior and the possibility to filter the tree is added by custom JavaScript code. The tree was created as a standalone component to make reuse possible; something that was done in Unipept 2.1 where the tree in multi-peptide analysis results page was replaced by this implementation.

jQuery UI is a set of user interface interactions, effects, and widgets built on top of the jQuery.

The table containing all genomes added to the analysis functions both as view and controller. D3 is used to generate the HTML table and to keep it synchronized with the internal status and order. Genomes can be added to the table by dragging them from the tree to the table with the help of jQuery UI. The same library is used to enable the reordering of genomes by drag and drop. The internal state of the application is kept in a single object containing all metadata that's shared with the pan-core graph.

The pan-core graph itself is an SVG element containing D3-generated content. The data shown in the graph is a combination of the metadata from the table and the pan and core data points that are calculated from the web worker. Next to visualizing the data, the graph can also be used to control the visualization. Drag and drop can be used to reorder and remove the genomes, and clicking on a data point shows a dialog containing more information and the option to download the corresponding data.

### Data flow

When adding a new genome to the analysis, it immediately gets added to the table with a *loading* status. The web worker then gets notified that a new genome was added and asynchronously requests the corresponding peptide data from the web server. The server answers the web worker with a gzip-compressed list of sorted integers in the JSON format. The web worker converts this list to a JavaScript array, stores it in an internal data object and starts calculating the union and intersection based on the previous state. The size of these two new sets are then sent back to the main JavaScript thread. In the main thread the new data is used to add new data points to the graph and the genome status is set to *done*. The flow for reordering the genomes is similar, except that no new data needs to be requested from the server.

After all queued genomes are loaded, the graph requests the web worker to load the unique peptide data. The web worker sends the server a list of all core peptides in the analysis, together with the taxon id's of all genomes in the analysis. The server calculates the LCA of these taxon id's and filters the list of peptides for those having the same LCA. The filtered list is then sent back to the web worker, which forwards the size of the received list to the main thread. The main thread then uses this information as new data points in the graph. The server needs the LCA for each of the submitted peptides to be able to filter on this value. Because the typical genome size is between 50 000 and 150 000 peptides, it's not feasible to calculate all LCAs on the fly. To solve this problem, the LCA for all peptides in the database is precomputed after constructing the database by making use of the LCA caching functionality introduced in Unipept 0.4. Unfortunately, this increases the already long time needed to process a new UniProt version with a few weeks.

### 5.2.5 Unipept version 2.1

Released on December 23, 2013.

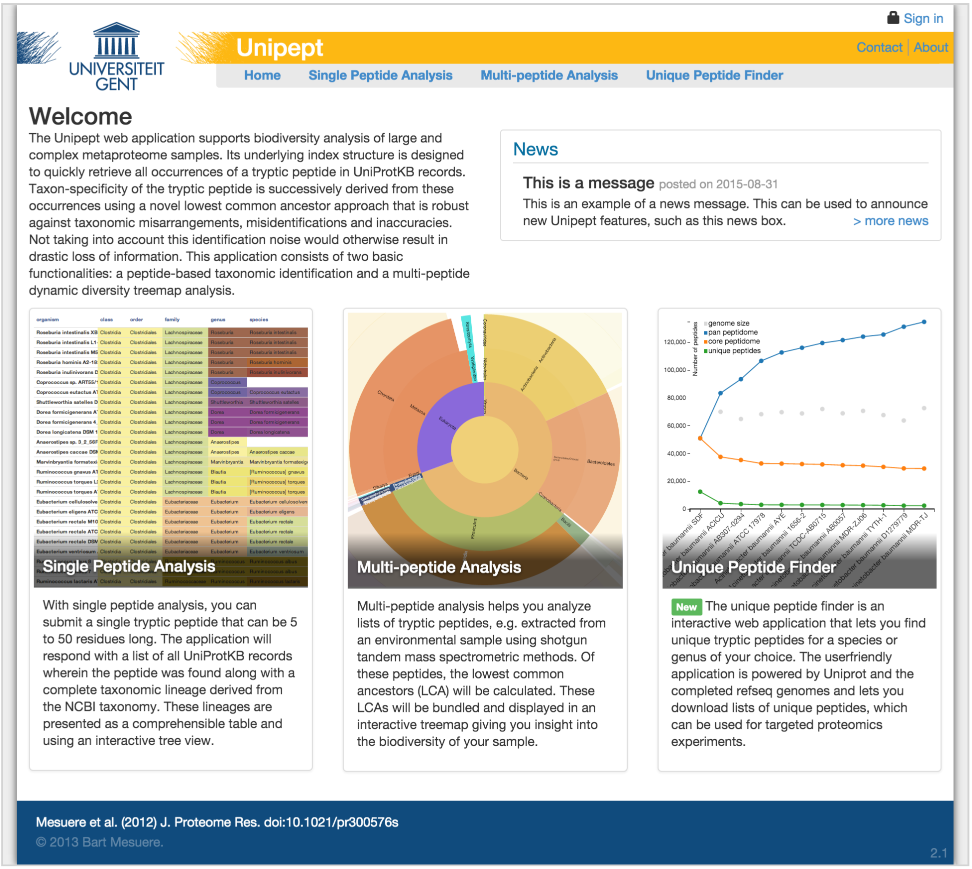
Unipept 2.1 was the smallest release in the Unipept 2 release series. Apart from the redesigned home page (Figure 5.14), it contained few new user-facing features. From a technical point of view, there are only two improvements worth mentioning: an improved file download flow and the refactoring of all unique peptide finder JavaScript code.

**Triggering file downloads**

Initiating a file download on a web page is harder than it seems, especially if the data is only present locally in the user's browser. Let's say we've got a JavaScript array containing a list of peptides and we want to let the user store these peptides on his computer. There is no straightforward way to write this data to a file on the user's file system using JavaScript only. The main reason for this is that in modern browsers JavaScript code gets executed in a sandbox which has no access to the local file system. The only reliable way to trigger a file download that works in all browsers is by doing a server roundtrip. With this strategy, the desired contents of the file is added to a hidden form on the page.

Chrome started work on a FileSystem API, but the project was abandoned because of the lack of interest from other browser vendors.

After triggering the form submission, the contents get sent to the server which then sends back the exact same data to the user. The server also sets the `disposition` header of the response to `attachment`. When the user's browser detects this header, it will trigger a file download dialog instead of rendering the response.



Welcome

The Unipept web application supports biodiversity analysis of large and complex metaproteome samples. Its underlying index structure is designed to quickly retrieve all occurrences of a tryptic peptide in UniProtKB records. Taxon-specificity of the tryptic peptide is successively derived from these occurrences using a novel lowest common ancestor approach that is robust against taxonomic misarrangements, misidentifications and inaccuracies. Not taking into account this identification noise would otherwise result in drastic loss of information. This application consists of two basic functionalities: a peptide-based taxonomic identification and a multi-peptide dynamic diversity treemap analysis.

News

This is a message posted on 2015-08-31

This is an example of a news message. This can be used to announce new Unipept features, such as this news box. > more news

Single Peptide Analysis

With single peptide analysis, you can submit a single tryptic peptide that can be 5 to 50 residues long. The application will respond with a list of all UniProtKB records wherein the peptide was found along with a complete taxonomic lineage derived from the NCBI taxonomy. These lineages are presented as a comprehensible table and using an interactive tree view.

Multi-peptide Analysis

Multi-peptide analysis helps you analyze lists of tryptic peptides, e.g. extracted from an environmental sample using shotgun tandem mass spectrometric methods. Of these peptides, the lowest common ancestors (LCA) will be calculated. These LCAs will be bundled and displayed in an interactive treemap giving you insight into the biodiversity of your sample.

Unique Peptide Finder

New The unique peptide finder is an interactive web application that lets you find unique tryptic peptides for a species or genus of your choice. The userfriendly application is powered by Uniprot and the completed refseq genomes and lets you download lists of unique peptides, which can be used for targeted proteomics experiments.

Mesuere et al. (2012) J. Proteome Res. doi:10.1021/pr300576s

© 2013 Bart Mesuere.

2.1

**Figure 5.14** The new home page layout as introduced in Unipept 2.1.

The downside of this approach is that sending all data to the server and back can take some time. According to Miller (1968), if a computer response takes longer than a second, some kind of confirmation or progress should be displayed to reassure the user. It is easy to display a notification that the file download is being prepared, but much harder to know when said notification can be removed. Because the download is triggered by an HTTP request, it's impossible to get direct status information on that request and thus on the status of the download. Our workaround sends a random number to the server along with the contents of the file. The server uses this number to set a cookie with a predefined name and the random number as its value. Meanwhile, the client-side JavaScript code keeps checking the list of cookies at a regular interval after the request was sent. Since the cookie only gets set when the server starts responding, we can remove the notification once the cookie with the correct value is detected.

A random number that is used for similar purposes is also called a nonce.

### JavaScript objects

To support the upcoming peptidome similarity feature, there was a need to refactor the Unipept peptide finder JavaScript code. The existing unique peptide finder code was located in a huge file that already contained over 1300 lines of code. Adding the peptidome similarity code to that same file was certainly not recommended. Until now, all internal functions were namespaced by wrapping them in a parent function that was executed on page load. Simply using another file would also create a new namespace, thus limiting the possible interactions between the two features. Because the two features share a lot of code, we embraced a more object oriented approach. JavaScript uses a prototypal inheritance model instead of a class-based model and has poor support for information hiding, which is why we designed our own class-like structure (Listing 5.1). This custom structure supports creating new objects with both public and private methods. All existing peptidome similarity code was refactored to adhere to the new structure and was split into separate files for each component. The possibility to create public and private methods allowed for a better design by defining a public API and hiding all internal methods.

**Listing 5.1** The class-like JavaScript structure used by Unipept.

```javascript
/**
 * Some comments about this class
 *
 * @param <String> name This is a string variable
 */
var constructMyClass = function constructMyClass(name) {
    /*************** Private variables ***************/

    // the object to which we will add all public methods
    var that = {};

    // a private variable
    var privateVariable = name;

    // a jquery variable
    var $jqueryVar = $("#someSelector");
    /*************** Private methods ***************/

    /**
     * Initializes method to set up the object.
     * Something constructor-like that gets called
     * at the end of the construct function
     */
    function init() {
        // set up some stuff
        privateMethod();
        that.publicMethod();
    }

    /**
     * This is a private method
     *
     * @param <Integer> i A magic number!
     */
    function privateMethod(i) {
        // Do something
    }
    /*************** Public methods ***************/

    /**
     * This is a public method that gets exported with
     * the created object
     */
    that.publicMethod = function publicMethod() {
        // do something
    };

    // initialize the object
    init();

    // return the object
    return that;
};
```

## 5.2.6 Unipept version 2.2

Released on June 6, 2014.

Unipept 2.2 introduced the peptidome clustering feature (Figure 5.15) and the possibility to add your own local genomes to the analysis. The user interface of the unique peptide finder was fine-tuned and a new coloring option was added to the sunburst visualization. The previous method selects the colors for the leaves by randomly picking them from a predefined list of colors and recursively determines matching colors for the parents. This results in a visually pleasing graphic, but the result is somewhat arbitrary. A small change in the dataset can result in a drastically different visual result. This is annoying when trying to compare the sunburst visualization of two datasets. The new coloring option calculates a hash of each of the taxon names in the dataset and uses that to deterministically assign a color to the corresponding leaves. This means that a certain taxonomic node will always have the same color, independent of the dataset.

The colors of the parents are calculated by taking the average of their first two children in de HSL color space.

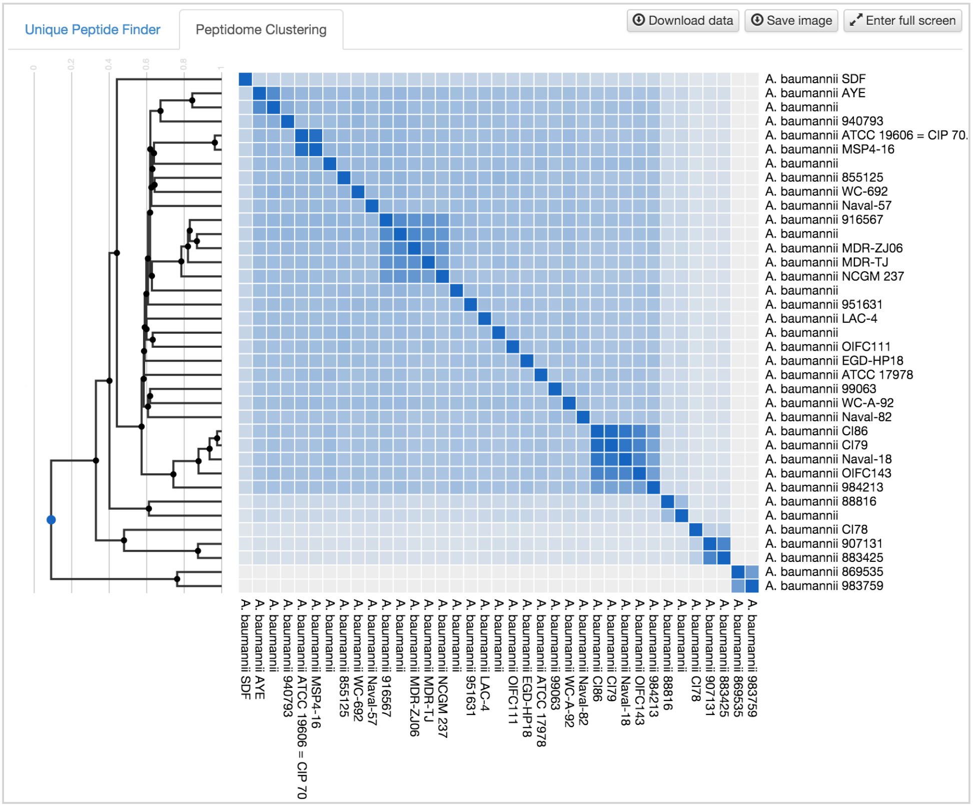


**Figure 5.15** The peptidome clustering feature as introduced in Unipept 2.2.

## Peptidome clustering

The peptidome clustering feature uses the peptidome contents of the genomes to calculate pairwise similarities. The similarity between two genomes is the number of peptides they have in common divided by the total number op peptides or, in other words, the size of the intersection divided by the size of the union of the two sets of peptides. These similarities are then used to perform a hierarchical clustering using UPGMA. The output of the clustering algorithm is then used to construct a phylogenetic tree. Both the tree and similarity matrix are rendered using D3 as a single visualization. This means that the branches of the phylogenetic tree are aligned with the squares in the similarity ma-

UPGMA stands for Unweighted Pair Group Method with Arithmetic Mean.

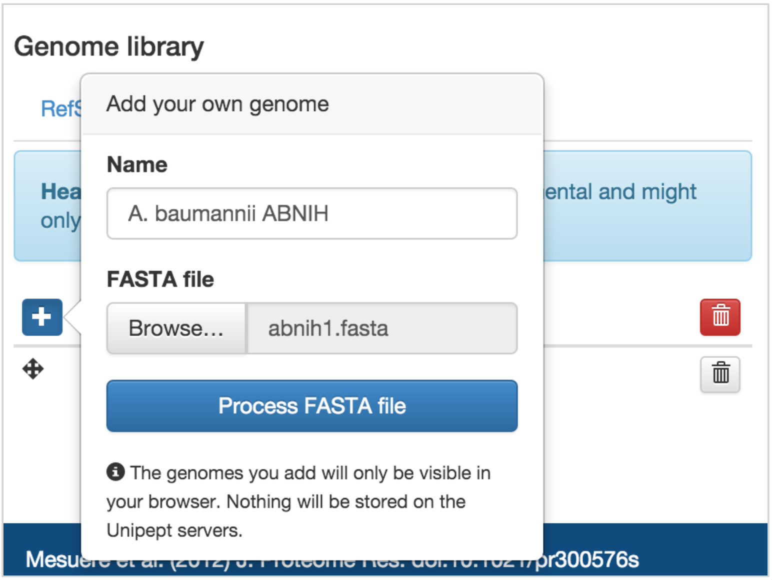
trix, and that clicking on a fork in the tree swaps both the branches and the corresponding rows and columns in the matrix.

Calculating the similarities is relatively fast (under 50 ms per pair), but because the number of pairs grows quadratically with the number of genomes in the analysis, the similarity calculation can't be done in the main JavaScript thread without interfering with the user experience. All necessary data is already present in the web worker of the unique peptide finder, so it was an obvious choice to add the code for the similarity calculation to the same worker. The similarity metric uses both the union and the intersection of the input sets, but because only the sizes are needed and not the actual contents, the calculation can be optimized. First, the intersection is calculated by iterating over the two sorted arrays containing the id's of the peptides in the input sets and counting the number of common peptides. The size of the union is then derived by taking the sum of the size of the input sets and subtracting the size of the intersection.

### My genomes

The second new big feature of Unipept 2.2 is the possibility to add genomes that are not (yet) in a public repository (Figure 5.16). This is done by allowing users to add a fasta file from their local hard drive containing all proteins of a genome. Because this is potentially confidential data, we store everything in the user's browser and not on the Unipept servers. A cross-browser method to achieve this is by using localStorage, a JavaScript API for persistent data storage. Unfortunately, storage is limited to 5 MB in most browsers, without a way to increase this. A later version of Unipept worked around this limitation by switching from localStorage to indexedDB if supported by the browser.

Internally, the genomes added by the user and the genomes derived from BioProjects are handled very similarly. The only difference is that instead of fetching the peptide contents from the server, localStorage is accessed. To do so efficiently, the locally stored genomes must be stored in the format as if they were returned from the server, meaning using their peptide id's. The conversion from a fasta file to a list of id's happens when adding the genome. To minimize the impact on the rest of the application, a separate web worker is used.



Name: A. baumannii ABNIH

FASTA file: abnih1.fasta

The genomes you add will only be visible in your browser. Nothing will be stored on the Unipept servers.

**Figure 5.16** The form to add locally stored genomes to the peptidome analysis.

### 5.2.7 Unipept version 2.3

Released on August 2, 2014.

The new treeview is discussed in more detail in Section 2.2.2.

Unipept 2.3 introduced a new data visualization to the multi-peptide analysis. Next to the existing sunburst and treemap, an interactive treeview (Figure 5.17) was added. The treeview is written in D3 and is a combination of a classic tree visualization and a Sankey diagram. This means that the size of the nodes and the width of the edges between them correspond to the number of matches for these nodes. Another change is the improved single peptide analysis page. External links to PRIDE and PeptideAtlas and cross references with EC numbers and GO terms were added to the table containing all UniProt matches.

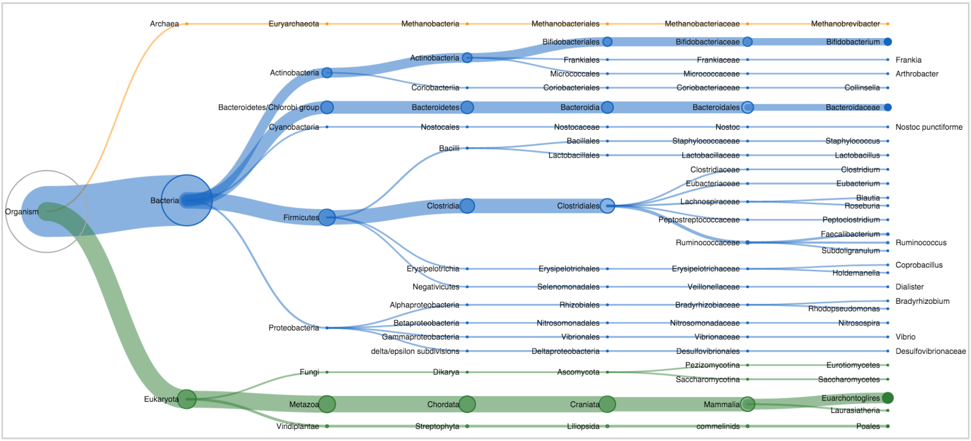
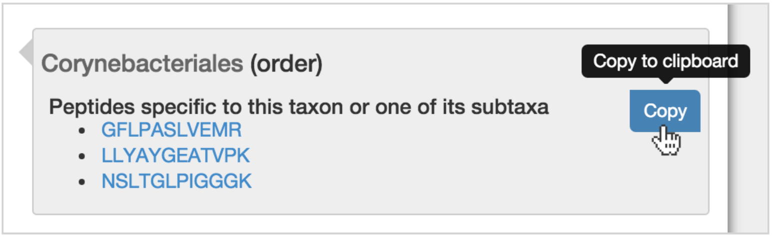


**Figure 5.17** The new treeview on the multi-peptide analysis page, added in Unipept 2.3.

## Copy to clipboard

An oft-requested feature was better support for data export out of Unipept. Although some pages already offered file exports, copying something to the clipboard is often more convenient. Because of the structure of the html pages it is not always possible to do this manually. For example, when the data is presented in a tabular form, it's impossible to manually select a single column and copy only that data. In this case, some sort of button to copy the contents of the column to the clipboard would solve the problem. Unfortunately, it's almost impossible to reliably access the clipboard across all browsers without external aid. We chose to use ZeroClipboard (http://zeroclipboard.org/), a library that provides clipboard access using an invisible Adobe Flash movie using a JavaScript interface. This library allows to programmatically copy any contents to the user's clipboard, enabling buttons such as the one in Figure 5.18.


Corynebacteriales (order)

Peptides specific to this taxon or one of its subtaxa
- GFLPASLVEMR
- LLYAYGEATVPK
- NSLTGLPIGGGK

**Figure 5.18** Example of the new copy to clipboard button using ZeroClipboard in Unipept 2.3.

## My genomes revisited

The initial version of My Genomes introduced in Unipept 2.2 was fully functional, but had some restrictions that limited its usefulness. The main obstacle was the limited storage space of 5 MB that's imposed by the use of localStorage. A solution for this limitation was to switch to IndexedDB, a standard for storing data in the browser using local databases. Because IndexedDB is not supported by all browsers, the old localStorage implementation is used as a fall back. The elimination of the size restrictions allowed us to fix a second problem with the initial implementation: the dependency on a single UniProt version. When parsing UniProt, peptides get assigned an id incrementally. This means that when parsing two versions of UniProt, a certain peptide will almost certainly get different id in the two versions. Because we only stored the sequence id's for the peptides in the user-submitted fasta file, the uploaded genomes became invalid when we updated the underlying UniProt database. To fix this, we now also store the original fasta file using IndexedDB. When the page detects that Unipept uses a newer version of UniProt, all stored genomes get updated.

IndexedDB is supported by Chrome, Firefox, and Internet Explorer (>=10). Unfortunately, the Safari implementation is riddled with bugs.

A consequence of unrestricted storage space is that a user potentially wants to add many custom genomes. Adding them one by one is a cumbersome process, which is why Unipept 2.3 added the option to add many files at once. In this case, the files are processed sequentially by the worker and a progress indicator is shown. The filenames are used as placeholder genome names and can later be edited.

### 5.2.8 Unipept version 2.4

Released on October 7, 2014.

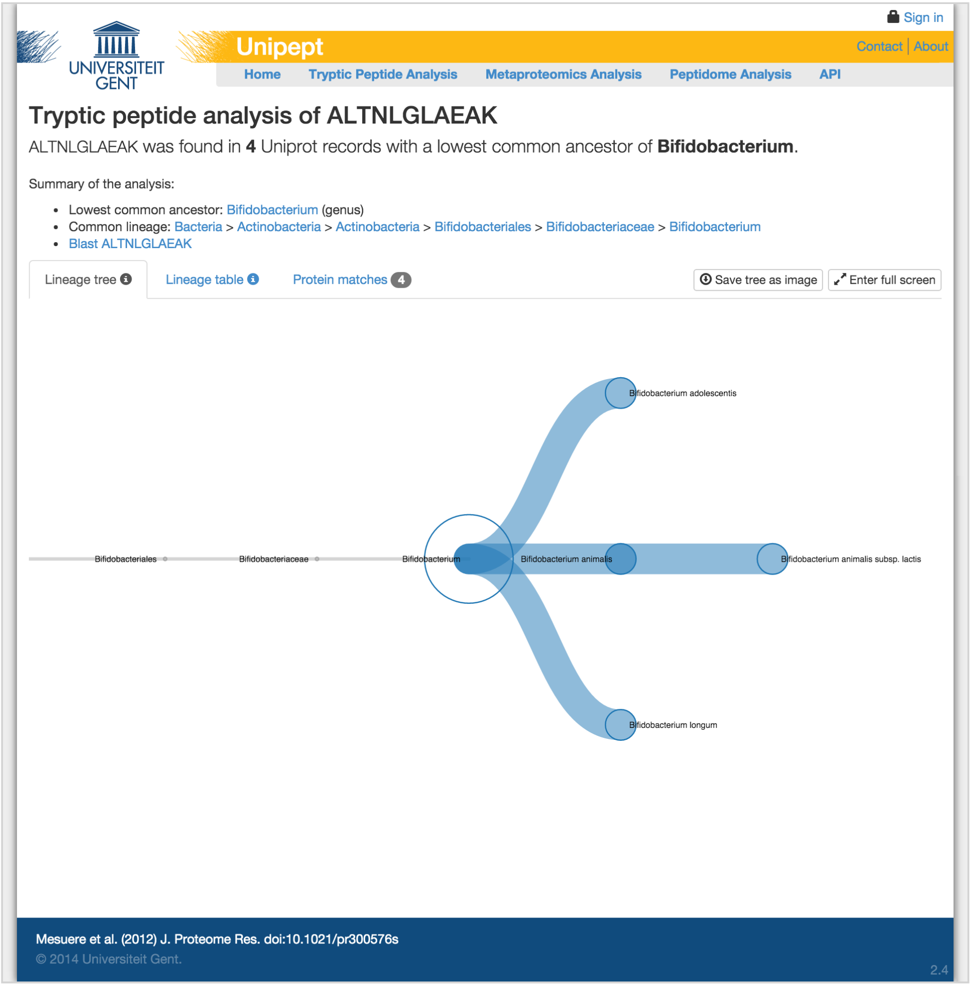
Unipept 2.4 adds an API with accompanying documentation to Unipept enabling the use of the LCA functionality in other tools and applications. Internally, all code was provided with tests and a rewrite of the UniProt processing pipeline was started with an improved LCA pre-computation step.

Multi-peptide analysis was renamed to metaproteomics analysis and a reset button was added to all visualizations. Single peptide analysis was renamed to tryptic peptide analysis and the results page was restructured. The old treeview was replaced by the one introduced in Unipept 2.3 (Figure 5.19).

#### API

More information about the API from a functional point of view can be found in Chapter 3.

To allow access to the Unipept peptide analysis features from within other applications, we created a simple RESTful API. The API contains five methods (`pept2prot`, `pept2taxa`, `pept2lca`, `taxa2lca`, and `taxonomy`) that can be accessed using HTTP POST or GET requests. The results are always returned in the JSON format. The API was designed with performance in mind, so by default, only a minimal number of fields are returned for each result. More information fields can be retrieved by explicitly asking for them when sending the request.



Tryptic peptide analysis of ALTNLGLAEAK

ALTNLGLAEAK was found in **4** Uniprot records with a lowest common ancestor of **Bifidobacterium**.

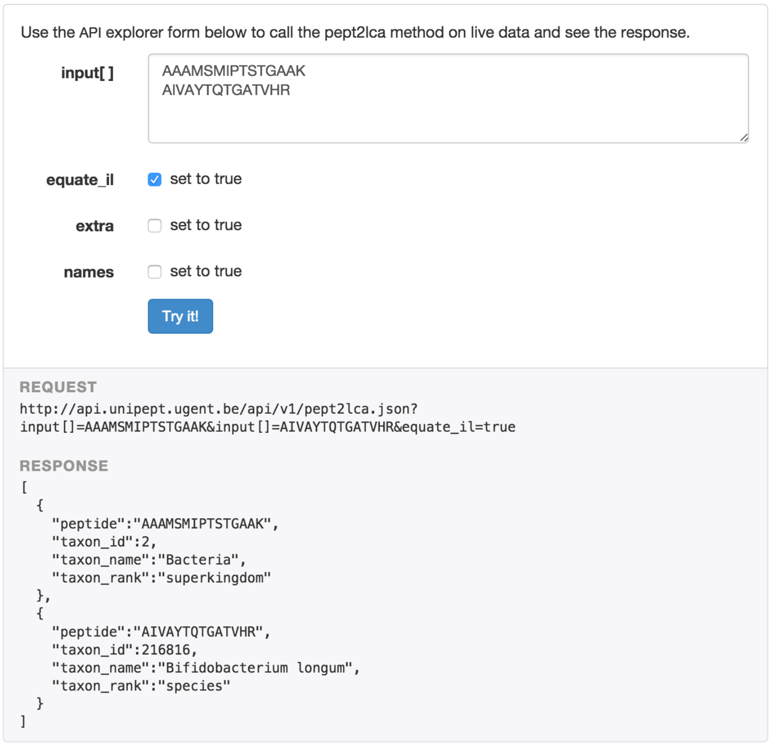
Summary of the analysis:

- Lowest common ancestor: Bifidobacterium (genus)
- Common lineage: Bacteria > Actinobacteria > Actinobacteria > Bifidobacteriales > Bifidobacteriaceae > Bifidobacterium
- Blast ALTNLGLAEAK

Mesuere et al. (2012) J. Proteome Res. doi:10.1021/pr300576s
© 2014 Universiteit Gent.

**Figure 5.19** The restructured tryptic peptide analysis page in Unipept 2.4.

The API comes with extensive documentation pages where every available method is explained in detailed. All possible input options and output fields are listed, along with several examples per method. Each documentation page also contains an interactive API explorer to compose requests and immediately see the results (Figure 5.20).


Use the API explorer form below to call the pept2lca method on live data and see the response.

| | |
|---|---|
| input[ ] | AAAMSMIPTSTGAAK<br>AIVAYTQTGATVHR |
| equate_il | ☑ set to true |
| extra | ☐ set to true |
| names | ☐ set to true |

REQUEST

```
http://api.unipept.ugent.be/api/v1/pept2lca.json?
input[]=AAAMSMIPTSTGAAK&input[]=AIVAYTQTGATVHR&equate_il=true
```

RESPONSE

```
[
  {
    "peptide":"AAAMSMIPTSTGAAK",
    "taxon_id":2,
    "taxon_name":"Bacteria",
    "taxon_rank":"superkingdom"
  },
  {
    "peptide":"AIVAYTQTGATVHR",
    "taxon_id":216816,
    "taxon_name":"Bifidobacterium longum",
    "taxon_rank":"species"
  }
]
```

**Figure 5.20** Screenshot of the API explorer, available on the documentation page of each of the Unipept API functions.

### Tests

Unipept aims to comply to as many software development best practices as possible. A major shortcoming to this was the lack of automated tests. Unipept 2.4 remedies this problem by adding over 900 tests and achieving a 100% test coverage for the Ruby on Rails part of the application. While this means that every single line of Ruby code is executed during testing, it doesn't imply that every line is actually tested.

The included framework is actually called TestUnit, a rails-specific flavor of Minitest.

Testing support is deeply integrated in Rails by the inclusion of a testing environment. This environment allows to specify separate settings for use during testing, for example a dedicated testing database. Furthermore, the testing framework Minitest is included since version 1.9. The

Minitest framework offers support for different types of tests and a way to actually run the tests.

All models are covered by unit tests, a way to test individual units of code. Because most models interact with the database, a separate test database is used. The data in this database is not actual data, but mock data specifically constructed for the tests. This mock data is specified in separate files called fixtures and is loaded into the database before every test is executed. The purpose of this is to create a known and fixed environment in which the tests can run so that the results are repeatable.

The code in the controllers is not as easily isolated as the code in the models. The controllers depend on many external elements and are therefore harder to test using unit tests. A different approach is to use integration tests. In integration tests, several modules are tested as a group, usually after testing the individual modules using unit tests. When performing an integration test on a controller, a web request is simulated and the output of the controller is compared with the expected output.

A third part of our testing strategy involves running RuboCop. RuboCop is a static code analyzer for Ruby that checks the code for offenses against the Ruby Style Guide. While it may seem that this only affects the appearance of the code, it also prevents errors due to sloppiness.

Having tests is of no use if you never run them, which is why we use Travis, a continuous integration service with GitHub integration. Whenever a commit gets pushed to GitHub, Travis fetches the code, runs all tests and posts the outcome of the tests in the pull request where the code was pushed. This helps guarantee that faulty code never gets deployed in production. The outcome of all tests is publicly available at https://travis-ci.org/unipept/unipept and can be used as a quality label for the code base.

**Computing the LCAs in Java**
The LCAs for all peptides in the database were precomputed by iterating over all peptides and calculating their LCA using the Ruby on Rails middleware. This is done for both the original peptide and the version where isoleucine and leucine were equated. Because the middleware fetched the information for each peptide separately, several database queries were executed per peptide, resulting in terrible performance. The total runtime of the LCA calculation using the middleware was three weeks for each of the two variants.

Unipept 2.4 includes a reimplementation of the LCA calculation code in Java. This code works directly in the generated tsv output files of the processing pipeline and has no interaction with the database. By keeping the lineage of all taxa in memory and making extensive use of the Java 8 Stream API, the new LCA calculation code is several orders of magnitude faster and runs in 30 minutes total for both variants. Profiling the Java code learns that almost all time is spent in reading and parsing the input files. This means that the algorithm can't be optimized any further. Another benefit is that because the code no longer makes use of the middleware and the database, it's much easier to integrate it in the data processing pipeline and to run it on a separate server.

### 5.2.9 Unipept version 2.5

Released on February 25, 2015.

Unipept 2.5 focused on improving the metaproteomics analysis visualizations. Until now, the sunburst and treeview graphs were created using D3.js and the treemap using the JavaScript InfoVis Toolkit (JIT, http://philogb.GitHub.io/jit/). This meant that the treemap was much more restricted in customizing the visualization and in adding new features. Another drawback is that the two sets of visualizations use a similar but slightly different input format. This means that the source data must be sent to the client twice, once for JIT and once for D3. Unipept 2.5 introduces a treemap replacement that is written using D3. This change allowed us to drop the JIT dependency from Unipept, unify the input format and pave the way for new treemap features.

By dropping JIT, the JavaScript code base was reduced by 500 KB.

**A D3 treemap**

Although D3 doesn't include ready made visualizations, it does have a few built-in layout algorithms. These algorithms encapsulate the process of converting input data to a set of shapes and/or positions. This output can then be used to create a graph. One of the built-in algorithms can be used to create a treemap from a hierarchical JSON object. When using this layout algorithm, a heuristic tries to recursively lay out rectangles associated with the input nodes. The algorithm used by D3 is superior to the one used by JIT, resulting in a better match between the size of the rectangles and the number of peptides associated with a node.

A second improvement is the addition of breadcrumbs at the top of the visualization (Figure 5.21). When a user zooms in by clicking on a node, these breadcrumbs show the path from the root till where the user zoomed in. This navigational aid allows the user to keep track of their location within the taxonomy tree. Clicking on one of the breadcrumbs results in zooming out to that level immediately, whereas previously, zooming out happened only a level at a time.

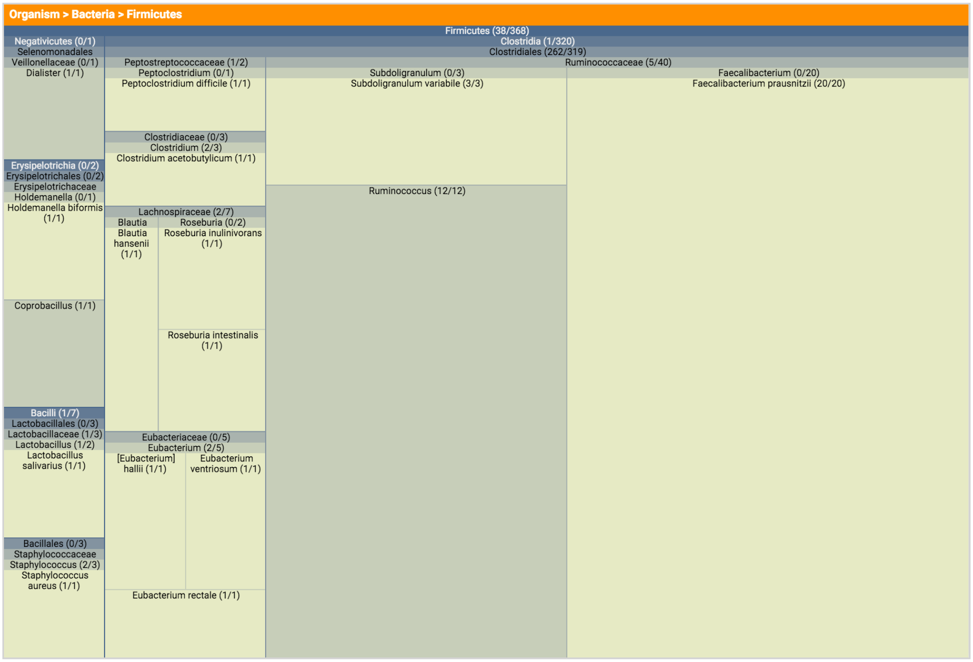


**Figure 5.21** The new D3-based treemap introduced in Unipept 2.5. The visualization uses an improved layout algorithm to determine the size of the rectangles and includes breadcrumbs as a navigation bar.

### Full screen mode

Apart from rewriting the treemap visualization using D3, the other visualizations were also refactored to make use of the JavaScript object style that was introduced in Unipept 2.1. As a consequence, the graphs became more modular and more flexible to work with. A direct result of this increased flexibility is a new full screen mode (Figure 5.22). Previously, full screen mode only showed a single visualization at a time without any other features. The new full screen mode lets users switch between visualizations and gives access to all export and display settings.

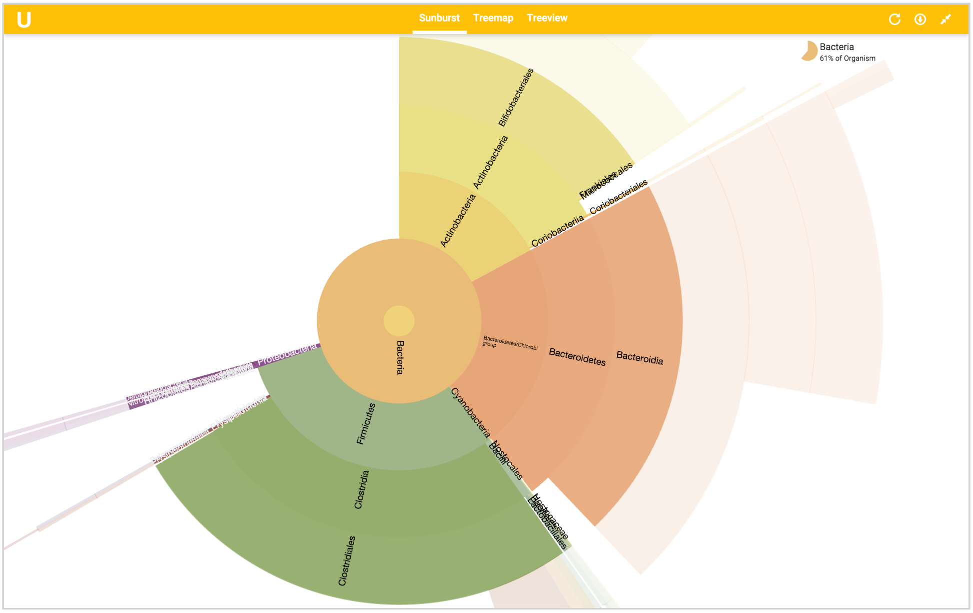


**Figure 5.22** Full screen mode of the Multi-peptide analysis visualizations demonstrating the new breadcrumbs of the sunburst graph.

The spaciousness of full screen mode and the treemap breadcrumbs provided the inspiration to also add breadcrumbs to the sunburst (Figure 5.23). A one-dimensional list of links works fine for the treemap, but doesn't fit the bold and circular sunburst. Instead, the circular theme was extended in the breadcrumbs by adding small pie charts showing the fraction of peptides at that level. The colors of the pie charts are the same as the ones used in the corresponding slices of the sunburst, creating an additional contextual link.

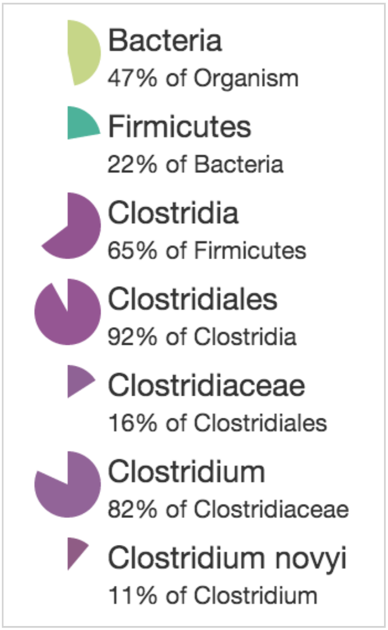
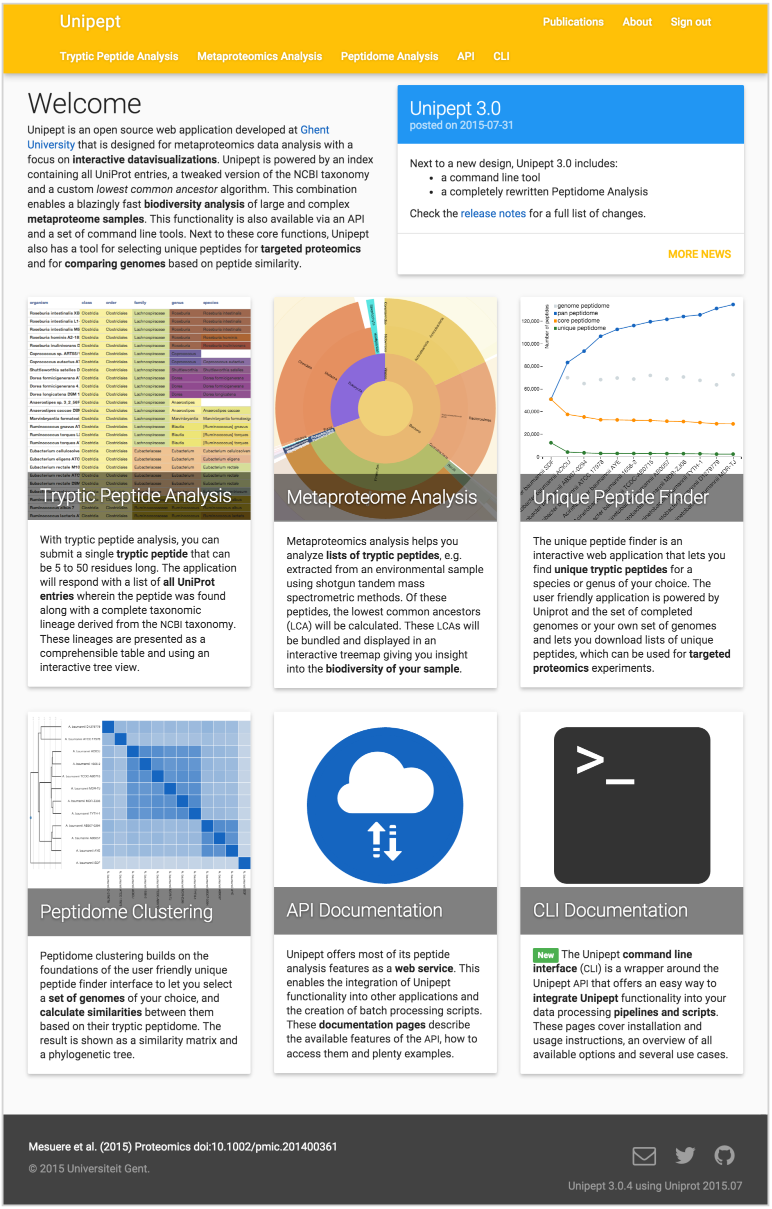


**Figure 5.23** Breadcrumbs of the sunburst visualization. Next to being a navigational aid, the miniature pie charts in the sunburst breadcrumbs also provide quantitative information.

### 5.2.10 Unipept version 3.0

Released on July 31, 2015.

Unipept 3.0 is the biggest Unipept release yet with major rewrites in both the back and front end and many new features. At the back end, the entire UniProt parsing pipeline was rewritten to make it several orders of magnitude faster. The entire website was redesigned based on Google's Material Design guidelines (http://www.google.com/design/spec/material-design). The unique peptide finder and peptidome clustering page was given a major update and the Unipept command line tools and accompanying documentation were released.



Unipept

Publications About Sign out

Tryptic Peptide Analysis Metaproteomics Analysis Peptidome Analysis API CLI

Welcome

Unipept is an open source web application developed at Ghent University that is designed for metaproteomics data analysis with a focus on **interactive datavisualizations**. Unipept is powered by an index containing all UniProt entries, a tweaked version of the NCBI taxonomy and a custom *lowest common ancestor* algorithm. This combination enables a blazingly fast **biodiversity analysis** of large and complex **metaproteome samples**. This functionality is also available via an API and a set of command line tools. Next to these core functions, Unipept also has a tool for selecting unique peptides for **targeted proteomics** and for **comparing genomes** based on peptide similarity.

Unipept 3.0
posted on 2015-07-31

Next to a new design, Unipept 3.0 includes:

- a command line tool
- a completely rewritten Peptidome Analysis

Check the release notes for a full list of changes.

MORE NEWS

Tryptic Peptide Analysis

With tryptic peptide analysis, you can submit a single **tryptic peptide** that can be 5 to 50 residues long. The application will respond with a list of **all UniProt entries** wherein the peptide was found along with a complete taxonomic lineage derived from the NCBI taxonomy. These lineages are presented as a comprehensible table and using an interactive tree view.

Metaproteome Analysis

Metaproteomics analysis helps you analyze **lists of tryptic peptides**, e.g. extracted from an environmental sample using shotgun tandem mass spectrometric methods. Of these peptides, the lowest common ancestors (LCA) will be calculated. These LCAs will be bundled and displayed in an interactive treemap giving you insight into the **biodiversity of your sample**.

Unique Peptide Finder

The unique peptide finder is an interactive web application that lets you find **unique tryptic peptides** for a species or genus of your choice. The user friendly application is powered by Uniprot and the set of completed genomes or your own set of genomes and lets you download lists of unique peptides, which can be used for **targeted proteomics** experiments.

Peptidome Clustering

Peptidome clustering builds on the foundations of the user friendly unique peptide finder interface to let you select a **set of genomes** of your choice, and **calculate similarities** between them based on their tryptic peptidome. The result is shown as a similarity matrix and a phylogenetic tree.

API Documentation

Unipept offers most of its peptide analysis features as a **web service**. This enables the integration of Unipept functionality into other applications and the creation of batch processing scripts. These **documentation pages** describe the available features of the API, how to access them and plenty examples.

CLI Documentation

New The Unipept **command line interface** (CLI) is a wrapper around the Unipept API that offers an easy way to **integrate Unipept** functionality into your data processing **pipelines and scripts**. These pages cover installation and usage instructions, an overview of all available options and several use cases.

Mesuere et al. (2015) Proteomics doi:10.1002/pmic.201400361
© 2015 Universiteit Gent.
Unipept 3.0.4 using Uniprot 2015.07

**Figure 5.24** Redesigned home page of Unipept 3.0, based on the Google Material Design guidelines.

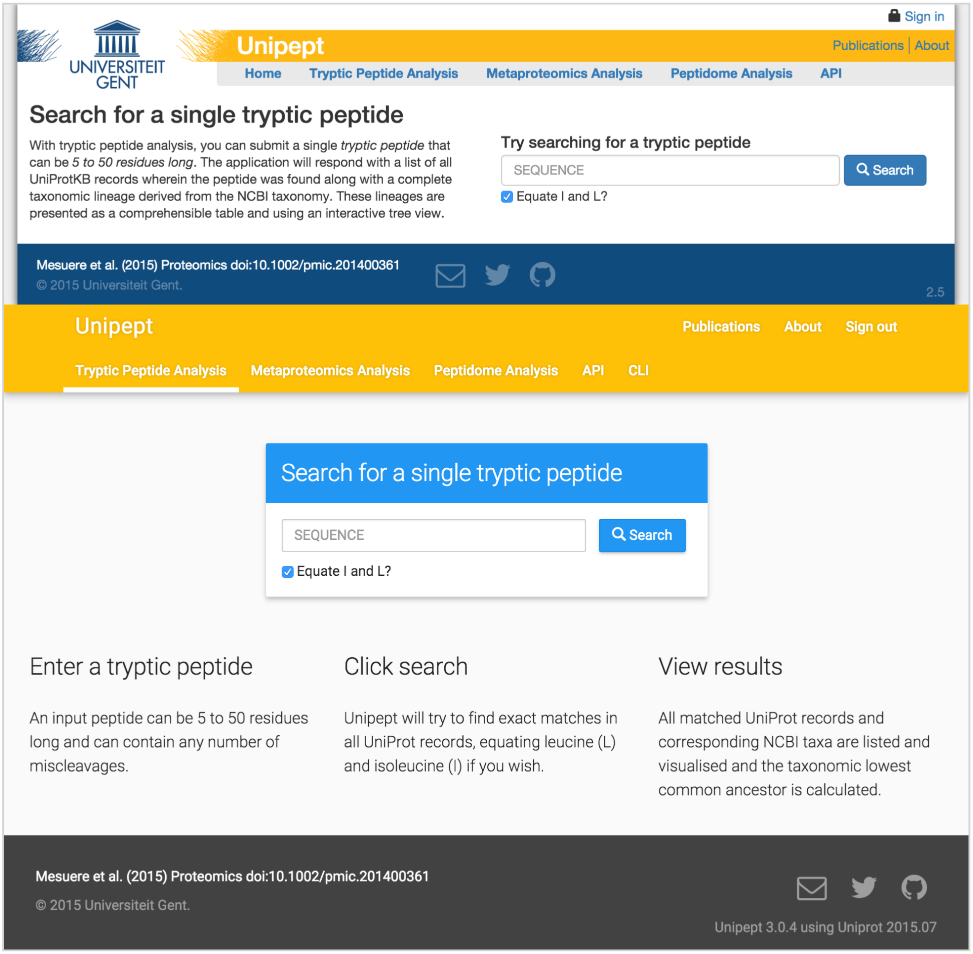
## Introducing Material Design

Until now, the Unipept page layout was based on the corporate design of Ghent University. This aging design was already heavily adapted to accommodate the needs of a modern web application. We chose to redesign the entire application based on Google's Material Design guidelines (Figure 5.24). These guidelines define a set of rules and principles to which applications must adhere to, leaving plenty of room for customization. These principles include using cards, depth, and padding to group information and using animations as a way to reinforce the user's actions.

The redesign was seized to rethink and improve all pages of the application. Special attention went to the removal of clutter and the simplification of the application. A prime example of this is the tryptic peptide analysis search page (Figure 5.25). The page now focusses on purpose of the page, being the search form, and the long help text was rewritten and split into three clear chunks.

### Proteome Analysis 2.0

Both the unique peptide finder and the peptidome similarity feature need a way to define complete genomes and to select all corresponding UniProt entries. In the past, BioProjects were used for this because each BioProject corresponded with a single sequenced genome. More recently however, this relation changed as a single BioProject ID can be used for a multi-species and multi-isolate project (Tatusova et al., 2015). As suggested by Tatusova et al. (2015), in Unipept 3.0 we started using Assemblies instead. By using the GenBank accession numbers associated with the assembly sequences, we can easily map them to UniProt entries for use in the application.


Sign in

Publications | About

Home Tryptic Peptide Analysis Metaproteomics Analysis Peptidome Analysis API

Search for a single tryptic peptide

With tryptic peptide analysis, you can submit a single *tryptic peptide* that can be *5 to 50 residues long*. The application will respond with a list of all UniProtKB records wherein the peptide was found along with a complete taxonomic lineage derived from the NCBI taxonomy. These lineages are presented as a comprehensible table and using an interactive tree view.

Try searching for a tryptic peptide

SEQUENCE Search

Equate I and L?

Mesuere et al. (2015) Proteomics doi:10.1002/pmic.201400361
© 2015 Universiteit Gent.

2.5

Unipept

Publications About Sign out

Tryptic Peptide Analysis Metaproteomics Analysis Peptidome Analysis API CLI

Search for a single tryptic peptide

SEQUENCE Search

Equate I and L?

Enter a tryptic peptide

An input peptide can be 5 to 50 residues long and can contain any number of miscleavages.

Click search

Unipept will try to find exact matches in all UniProt records, equating leucine (L) and isoleucine (I) if you wish.

View results

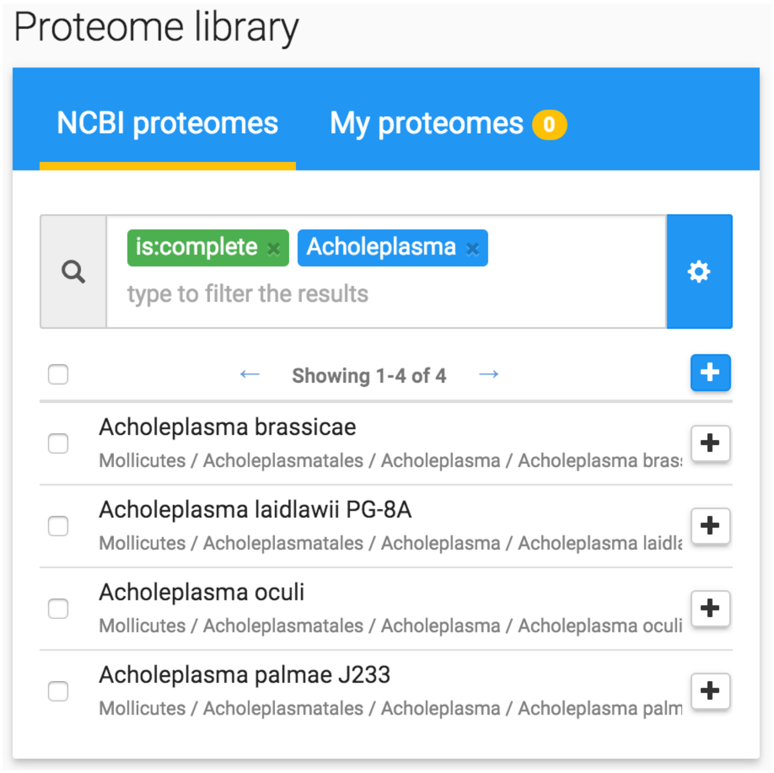
All matched UniProt records and corresponding NCBI taxa are listed and visualised and the taxonomic lowest common ancestor is calculated.

Mesuere et al. (2015) Proteomics doi:10.1002/pmic.201400361
© 2015 Universiteit Gent.

Unipept 3.0.4 using Uniprot 2015.07

**Figure 5.25** Compilation showing the old (Unipept 2.5) and redesigned (Unipept 3.0) tryptic peptide analysis page. The top navigation bar now shows the current page, the search form is front and center and the copy was improved.

By using the assemblies, which also include incomplete genomes, the list of available genomes grew considerably to over 7 500. This had performance implications on the tree that was used to add genomes to the analysis. The initial rendering time of the selection tree grew to over 10 seconds, which is unacceptable. A new solution was found in using a filterable list, showing only 50 results per page (Figure 5.26). The improved search settings include options to target specific taxonomic ranks, filter on assembly level (e.g., complete or scaffold), filter on complete or partial genomes and to only display type strains.



Proteome library

Acholeplasma brassicae
Mollicutes / Acholeplasmatales / Acholeplasma / Acholeplasma bras

Acholeplasma laidlawii PG-8A
Mollicutes / Acholeplasmatales / Acholeplasma / Acholeplasma laidl

Acholeplasma oculi
Mollicutes / Acholeplasmatales / Acholeplasma / Acholeplasma oculi

Acholeplasma palmae J233
Mollicutes / Acholeplasmatales / Acholeplasma / Acholeplasma palm

**Figure 5.26** The proteome library showing a new way of selecting proteomes by using a filterable list instead of hierarchical lists. All complete genomes of the *Acholeplasma* genus are shown.

The idea of adding more functionality to the full screen mode of the metaproteomics analysis was also applied on the proteome analysis. While in full screen, users can now switch between the unique peptide finder and the peptidome similarity and can even add new proteomes to the analysis. Another improvement to the proteome analysis is the automatic recovery of the analysis state of the previous visit to the page. To eliminate the risk of getting caught in a restoration/crash loop, the same state is only restored once.

**Modern JavaScript**

While rewriting big parts of the peptidome analysis feature, the opportunity was seized to introduce a few new JavaScript features in the Unipept code base. These features are all part of ECMAScript 2015, a new JavaScript language specification that was standardized in June 2015. Because not all browsers support these features yet, Unipept only uses functionality that can be mimicked by using polyfills. A polyfill is a piece of code that can be used to replicate a functionality that the browser doesn't support natively. An example of an ES2015 feature that is used in Unipept 3.0 are the new Map and Set data structures.

ES2015 was previously called ECMAScript 6 or ES6.

A second example of a new ES2015 feature that is used in Unipept are Promises. Traditionally, JavaScript programmers make liberal use of callbacks to handle the results of asynchronous functions. Callbacks are functions that are supplied by the programmer when calling an, often asynchronous, function and are typically executed by that function when the result is available. The problem with this approach is that after calling the initial function, the programmer has lost control over the subsequent program flow. Additionally, callbacks seem to be contagious and have the nature to propagate through the source code, resulting in a phenomenon called *callback hell*. Most of these problems can be solved using Promises, a way to refer to the result of operations that aren't available yet, but will be in the near future.

This problem is called inversion of control.

All JavaScript code in Unipept that used callbacks for asynchronous functions was rewritten to make use of Promises. The peptidome analysis page makes heavy use of JavaScript Workers and therefore callbacks. The switch to Promises drastically reduced the complexity of this code and made the program flow much more transparent.

### The Unipept command line tools

After an API was added in Unipept 2.4, work began on a Unipept command line interface (CLI) tool to easily integrate Unipept functionality in batch processing scripts. A major part in creating the tool was providing ample documentation about its usage. This documentation was included in Unipept 3.0 and was created in analogy to the API documentation. Each of the included commands have their own documentation page listing all available input and output options and a few examples. Additionally, the documentation includes two detailed case studies: the taxonomic analysis of a tryptic peptide and of a metaproteomics dataset. More information about the command line tool itself can be found in Section 3.2.

### A file-based UniProt parser

In Unipept 2.4, the LCA calculation was reimplemented to use a file-based approach instead of the database. This resulted in such a huge performance improvement that the entire processing pipeline was rewritten using the same ideas. Instead of directly writing the data to the database, the individual parsing steps now generated tab separated value (TSV) files that can easily be imported into the database afterwards. The entire pipeline can be executed by running a single makefile that runs and combines the various processing steps.

For all but one table, this was a straightforward conversion. Because these tables are only written to during processing, it required minimal changes to the code. The `sequences` table is the exception here because we both read from and write to this table in the processing step. More specifically, before a sequence is added to the "database" we first want to check if it was not previously added, and if this is the case, with which identifier. To solve this problem, we used

Berkeley DB to store the sequence-identifier mapping during processing. Berkeley DB is a high-performance software library to store key-value pairs. Berkeley DB will keep as much data in memory as possible and will use the hard drive as a fallback option when memory is exhausted.

The performance improvements of this reimplementation are similar to those achieved with the new LCA approach. Parsing the entire UniProt database can now be done in under 12 hours compared to over four weeks with the old approach. This allows us to provide more timely updates to the Unipept database. Another attempt to reduce the number of lookup operations by making use of bloom filters did not result in any significant performance improvements.

### UniProt redundancy removal

Over the course of 2014, the size of the UniProt database more than doubled from 45 million entries to over 90 million (Figure 5.27). The UniProt team attributes this exponential growth to the high number of redundant bacterial genomes. As an example, they show that UniProt contains 4 000 proteomes for *Staphylococcus aureus*, accounting for 10 million UniProt entries. In order to deal with this drastic data growth, they implemented a redundancy removal procedure. The procedure removes redundant UniProt entries by identifying similar bacterial genomes. As a result, the size of the UniProt database decreased from 92 million entries to just 47 million.

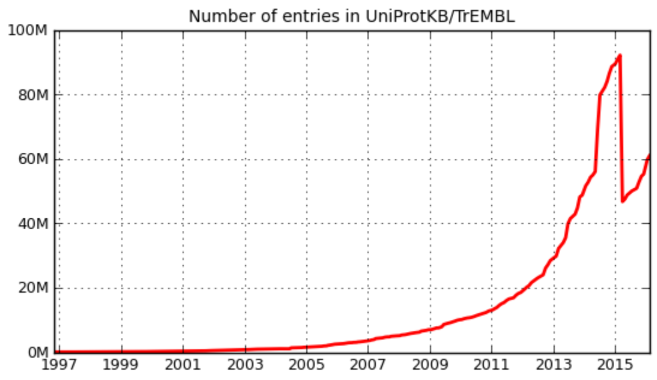


**Figure 5.27** A graph showing the growth rate of the UniProt database. Its size was approaching 100 million entries in early 2015 after which a redundancy removal was implemented.

Although this redundancy removal has nothing to do with changes in Unipept 3.0, it was the first Unipept version that was impacted by this reduction in source data. As expected and verified by spot checks, the impact on the Tryptic Peptide Analysis and Metaproteomics Analysis was minimal. Because only redundant entries were removed, there is enough data left to reliably assign peptides to taxa. The Unique Peptide Finder and Peptidome Clustering on the other hand, were seriously affected by the change in UniProt. To properly function, these features rely on the presence of multiple genomes/proteomes per species. The more data is available, the better these features work. The removal of half of the database thus had a severe negative impact on the usefulness of the peptidome analysis. We therefore regret the decision of the UniProt team to throw away half of their data.

### 5.2.11 Unipept version 3.1

Released on December 11, 2015.

To counter the problems caused by the UniProt redundancy removal, we once again switched our data source for the Peptidome Analysis in Unipept 3.1. Using the assembly data proved increasingly unreliable and caused several incomplete proteomes to show up in our list of proteomes. In our search for an alternative, we reevaluated the use of the proteomes as published on the UniProt website. We considered using these proteomes before, because they are guaranteed to be complete. The reason for not using them earlier was that they only contained few proteomes per species, something essential for the Peptidome Analysis. This issue was remedied in more recent UniProt releases when more proteomes became available. An overview of the number of available proteomes for a few key species can be seen in Table 5.1. Because the previous rewrite was done with flexibility of the underlying data in mind, the change from assemblies to UniProt proteomes required relatively few code changes.

**Table 5.1** Number of proteomes for which Unipept has data for both the old (complete assemblies) and the new (UniProt complete non-redundant proteomes) data source. Both the total number of proteomes and the number of proteomes for five key species are shown. The UniProt numbers are taken from the 2015.11 release.

| ORGANISM | NUMBER OF ASSEMBLIES | NUMBER OF UNIPROT PROTEOMES |
|---|---:|---:|
| *Acinetobacter baumannii* | 16 | 31 |
| *Escherichia coli* | 61 | 151 |
| *Staphylococcus aureus* | 22 | 20 |
| *Bacillus cereus* | 16 | 107 |
| *Lactococcus lactis* | 13 | 18 |
| **Total** | 7 715 | 10 638 |

## Delta encoding

A second focus of Unipept 3.1 was performance improvements of the Peptidome Analysis. One of the things that happen constantly when using this feature is downloading

More information in these id's can be found in Section 5.2.4.

proteomes from the Unipept server. Although sending peptide id's instead of the full sequences already saves us some space and thus download time, we can do better. When sending the id's to the browser, these integers are actually sent as text (i.e., as separate digits) and not as a number. This means that for example the number 123 456 takes up six times more space than the number 3. Sending shorter, lower numbers would thus mean sending less data.

One way to achieve this is by performing a delta encoding on the id's before sending them. With delta encoding, instead of sending the numbers themselves, we send the arithmetic difference between the actual value and the value of the previous element in the list. For example, the list `[101,102,104,105,110,111]` would get sent as `[101,1,2,1,5,1]`, a 40% reduction in characters. Because we don't have any negative id's, this results in lower numbers by definition and because our initial list is sorted, we achieve a maximal reduction in size. The client can easily reconstruct the original list by simply incrementing a temporary variable with the received values.

Applying delta encoding on the sequence id's had a few other unexpected benefits. When data gets sent from a web server to a browser, most of the time the data is compressed using gzip compression. This compression technique works best if there are a lot of repeating values in the data, something that is certainly the case with our delta-encoded id's. Especially when taking into account that because of the way the id's are assigned during database construction, there's a high chance of having sequential id's in a proteome. As can be seen in Table 5.2, applying delta encoding reduces the proteome size (and download time) by a factor of three. Consequently, the space needed to store a cached version of our proteomes is also reduced by the same amount, just as the time the browser needs to convert the received (JSON) response to a list of integers.

**Table 5.2** The size of a proteome of *Brassica napus* (rapeseed), *Homo sapiens* (human), *Escherichia coli* and *Acinetobacter baumannii* after various encoding and compression steps. *Sequence size* shows the size of the proteome if the actual sequences are used, *id size* if the sequences are represented by their id (in textual form), *encoded id size* if the sequence id's are delta encoded and *compressed size* after applying gzip compression on the delta encoded result.

| ORGANISM | # PEPTIDES | SEQUENCE SIZE | ID SIZE | Δ ENCODED SIZE | COMPRESSED SIZE |
|---|---|---|---|---|---|
| *B. napus* | 1 257 794 | 20.7 MB | 11.8 MB | 3.00 MB | 835 KB |
| *H. sapiens* | 722 407 | 11.5 MB | 5.8 MB | 1.60 MB | 395 KB |
| *E. coli* | 84 126 | 1.4 MB | 0.7 MB | 0.21 MB | 63 KB |
| *A. baumannii* | 69 723 | 1.1 MB | 0.6 MB | 0.19 MB | 68 KB |

# Chapter 6

# The future of Unipept

This chapter explores possible future extensions to the Unipept platform. Four of them are discussed in detail: the possibility to use Unipept for shotgun metagenomics datasets, the addition of a functional analysis next to the current diversity analysis, the addition of a statistically-sound comparative analysis and improved filtering of unique peptides for targeted metaproteomics.

## 6.1 From metagenomics to metaproteomics (and back again)

Consider the problem of taxonomically assigning a DNA read resulting from a shotgun metagenomics experiment. The initial step in this process, shared by all current metagenomics analysis tools, is to perform inexact matching (e.g., Blast (Altschul et al., 1990), BLAT (Kent, 2002), USearch (Edgar, 2010)) of the DNA read against a reference database and to retain only those hits that are considered sufficiently similar. Some tools only consider the best match and assign the DNA read based on the taxonomic annotation of that match in the reference database (Seshadri et al., 2007). Other tools consider multiple matches (e.g., a fixed number of best matches or all matches that meet certain cutoff criteria) and use an aggregation strategy to reduce the taxonomic annotation of those matches in the reference database into a consensus taxonomic assignment of the DNA read (Huson et al., 2007; Meyer et al., 2008; Luo, Rodriguez-R, and Konstantinidis, 2014). Pep2Pro (Askenazi, Marto, and Linial, 2010), for example, only makes a taxonomic assignment if all retained matches share the same taxonomic annotation in the reference database. This is particularly bad since it strongly depends on the sampling bias of the reference database and favors dominant taxa.

A far better approach that is generally recommended in review papers, is to compute the lowest common ancestor (LCA) of all retained matches (Bazinet and Cummings, 2012; Mande, Mohammed, and Ghosh, 2012). However, such a strategy should be implemented carefully, as the

NCBI Taxonomy for example contains a large number of "false" taxa (strain-level assignments; Federhen et al. (2014)) that should be discarded before applying the LCA procedure, and the procedure should also be robust against "false" taxonomic annotations in the reference databases. To avoid "false" reference sequences, databases are often compiled from a list of whole-genome sequences, reducing the taxonomic identification space to only those taxa whose complete genome was sequenced, or from a list of quality controlled 16S rRNA sequences, ignoring the majority of DNA reads that do not contain 16S rRNA fragments and reducing the taxonomic identification space to the prokaryotes (Hunter et al., 2014).

## A shotgun metagenomics pipeline

The Unipept platform could be extended with a shotgun metagenomics analysis pipeline that solves the taxonomic assignment of DNA reads following a radically different approach. Because Unipept was originally designed for shotgun metaproteomics analysis, the metagenomics problem could be translated into a series of metaproteomics problems. These can be solved using existing Unipept functionality and the results can be mapped back into the metagenomics context.

Figure 6.1 outlines how this approach can be implemented. Instead of directly matching a DNA read against the sequences in a reference database, we start by predicting protein coding genes on the DNA read. This can be done using software packages like FragGeneScan (Rho, Tang, and Ye, 2010) or its multi-threaded variant FragGeneScan+ (Kim et al., 2015), which is particularly interesting since it makes use of a hidden Markov model that allows to i) find genes that are only partially covered in the DNA read, ii) correct read errors induced by current NGS technologies and iii) predict the translation table that needs to be used to translate the nucleotide sequence into a protein sequence.

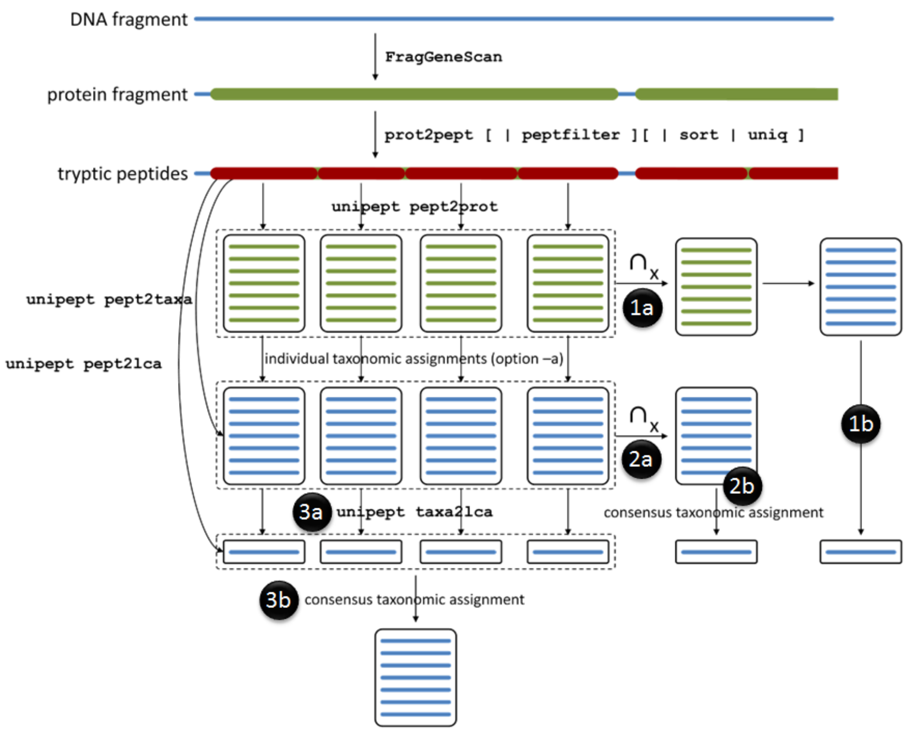

DNA fragment
FragGeneScan
protein fragment
prot2pept [ | peptfilter ][ | sort | uniq ]
tryptic peptides
unipept pept2prot
unipept pept2taxa
unipept pept2lca
1a
individual taxonomic assignments (option –a)
1b
2a
2b
consensus taxonomic assignment
3a
unipept taxa2lca
3b
consensus taxonomic assignment

**Figure 6.1** General outline of the strategies for taxonomic identification of a DNA read by directly predicting gene fragments on the DNA read, breaking up the partial proteins into tryptic peptides and finding all proteins having exact matches with the tryptic peptides. In a next step, a consensus taxonomic assignment can be computed from (1) the proteins having multiple peptide hits, (2) the taxa having multiple peptide hits, or (3) the taxonomic assignments for the individual peptides.

Assembly of partial peptide fragments into more complete protein sequences using the LCA short peptide assembler (Yang and Yooseph, 2013) is an optional post-processing step, which is in contrast with more traditional approaches that start with assembly of DNA reads, followed by gene prediction (Nielsen et al., 2014). This approach is far more accurate than using nucleotide assembly-based strategies followed by gene prediction. Following this strategy, we essentially transform the metagenomics problem into a metaproteomics problem that can already be solved by Unipept.

The predicted (partial) protein sequences are digested *in silico* into a series of tryptic peptides that are individually identified taxonomically using the Unipept LCA procedure (Figure 3.4; Figure 6.1, 3a). Unipept uses exact matching to map peptides to proteins in UniProt, which is not only faster than the inexact matching procedures that are used traditionally but also avoids the need for cutoff criteria: the LCA is computed based on all exact matches.

Although it might seem that the Unipept approach will be less accurate because inexact (DNA) matching is less stringent than exact (protein) matching, Tanca et al. (2013) have shown in a controlled experiment that Unipept has three to five times more correct identifications compared to MEGAN (Huson et al. (2007); which also implements an LCA approach but based on Blast matches), with more than half the number of incorrect identifications. In addition, since Unipept precomputes the LCA for all tryptic peptides extracted from the UniProt database, this step in the identification process is extremely fast.

### Aggregation

Although the taxonomic identification process could stop at this point by simply pooling all individual peptide-based identifications for all DNA reads in a shotgun metagenomics data set, this approach would ignore an important piece of information: all tryptic peptides derived from the same DNA read come from the same organism. This is where metagenomics has an advantage over metaproteomics, because we can perform an additional aggregation step that bundles the individual peptide-based identifications into a single consensus identification for the DNA read

(Figure 6.1, 3b), which can be more accurate (e.g., less specific identifications are overruled by more specific identifications) and more robust against incorrect identifications (e.g., less supported identifications are pruned). Such an aggregation essentially transforms the metaproteomics identifications back into a consensus metagenomics identification. The individual peptide-based identifications could for example be aggregated by again computing their LCA (note that these LCA computations cannot be precomputed). Wood and Salzberg (2014) propose another aggregation technique based on determining the maximal root-to-leaf (RTL) path of all individual identifications in the taxonomic tree.

Both of the above approaches are based on the idea that aggregation at this step should reduce a series of individual taxonomic assignments into a single taxonomic assignment. However, it is statistically more sensitive to postpone such a crisp reduction until later stages (where the overall biodiversity distribution of the whole sample is analyzed and visualized) and to apply more fuzzy reductions at this stage that represent the aggregated biodiversity as a weighted hierarchical distribution.

One possible approach is illustrated in Figure 6.2. Instead of reducing the weighted hierarchical distribution formed by the individual taxonomic assignments to a single taxonomic node (for example by taking the lowest node with weight 1 in the LCA approach, or selecting the leaf node with the highest weight in the RTL approach), we may represent the taxonomic assignment of a single DNA read (or protein) as a weighted tree. These weighted trees can then be aggregated for all DNA reads sequenced from the sample by adding the weights of the corresponding nodes in the hierarchy and divide these weights by the number of hierarchies merged (i.e., normalization over all proteins or DNA reads). This essentially computes the root-to-node (RTN) path for every node in the tree (in contrast to computing the path only for the leafs with the RTL approach). As an end result, the biodiversity distribution of a particular sample is represented as a weighted tree instead of a count table (or tree), which is just a generalization that can equally well be analyzed and visualized. A proof-of-concept of this workflow has already been implemented in collaboration with the EBI Metagenomics team, showing very accurate identifications that are not restricted to the prokaryotes as with most of the other metagenomics pipelines. This basic implementation already sequentially processes 12 million reads in under an hour.

Starting from a DNA read that is broken into a series of tryptic peptides, another aggregation approach is to map each peptide onto its matching proteins (Figure 6.1, 1a) and aggregate the taxonomic annotation only from those proteins that match all or a minimal number of the peptides from the query protein (Figure 6.1, 1b), e.g., using the Unipept LCA approach. A variant thereof is to map each peptide onto the taxonomic annotations of all its matching proteins (Figure 6.1, 2a) and only aggregate the taxonomic annotations that were matches for all or a minimal number of the peptides from the query protein or DNA read (Figure 6.1, 2b), e.g., using the Unipept LCA approach. This will probably lead to more accurate results compared to the previous approaches, but at the cost of a much longer computing time because some universal peptides will be found in large numbers of proteins and/or taxa.

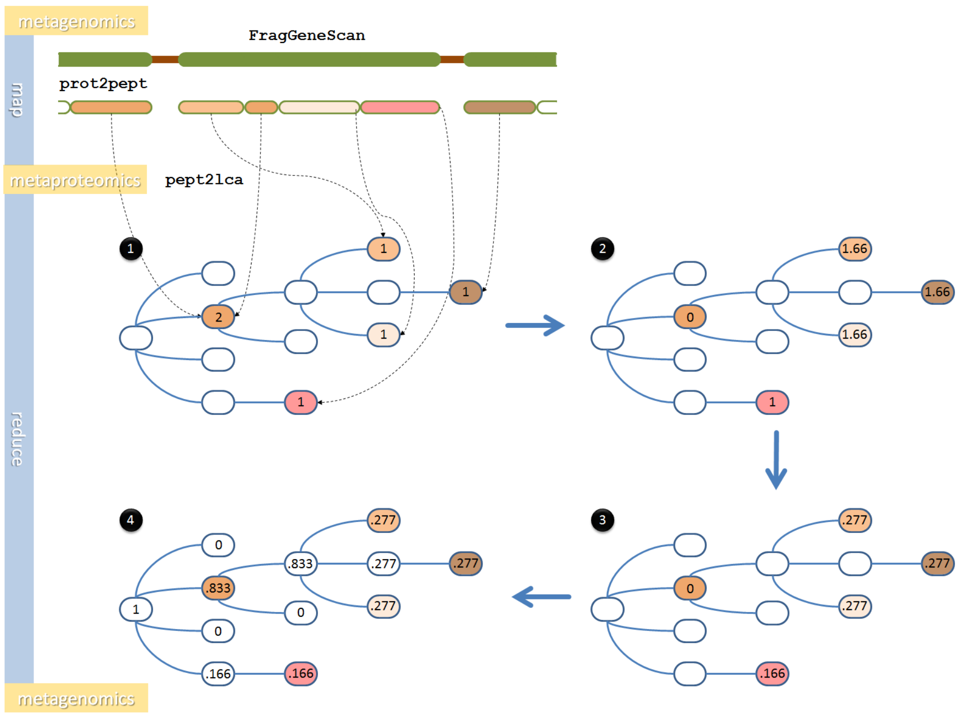


**Figure 6.2** Weighted hierarchical distribution algorithm: (1) assign (normalized) weights to taxa based on number of individual (peptide) identifications, (2) take into account the hierarchical structure by distributing the weight of a parent node among the identified leaf nodes of its subtree (optional); distribution of parent weights may or may not take into account weights already assigned to leafs, (3) normalize weights across the entire tree (each protein identification is assigned the same weight), (4) take into account the hierarchical structure by computing parent weights as the sum of their child weights (optional).

### k-mer index

Apart from *in silico* splitting a (partial) protein into (non-overlapping) tryptic peptides, the protein may also be split into (overlapping) k-mers. This approach has been introduced in KRAKEN (Wood and Salzberg, 2014). The same aggregation strategies that were discussed for peptide-based identifications can also be applied onto the k-mers. Note however that KRAKEN introduces bias by not correcting for identifications derived from overlapping k-mers, which can be easily remedied by normalizing the identifications per protein residue. It is expected that identification based on overlapping k-mers will increase accuracy compared to peptide-based identification (which is only motivated in a metagenomics context by the fact that Unipept already has fast peptide-based indexes for usage in a metaproteomics context), but at the cost of a higher memory footprint to store the index and a performance penalty because the number of overlapping k-mers is higher than the number of non-overlapping tryptic peptides.

### Preliminary results

A prototype based on the Unipept command line tools is already available. The prototype takes two fastq files containing paired-end reads as input and tries to map a taxon to each of the reads as output. The pipeline consists of four basic steps.

1. Merge the reads of the two fastq files into a single fasta file. This is done using basic Unix command line tools.
2. Run FragGeneScan+ to convert the reads into protein fragments.
3. Split the protein fragments into tryptic peptides and calculate the LCA for each of the peptides using the Unipept command line tools.
4. Aggregate the results of the individual peptides per read into a single identification for each read using a custom python script.

As a first test, we ran the pipeline on simulated reads from a mix of four organisms (*Escherichia coli*, *Plasmodium falciparum*, *Shigella dysenteriae*, and *Human immunodeficiency virus*). Our prototype was able to process 2 times 9 million 100 base pair (bp) reads in 15 minutes, or 1.2 million 100 bp reads per minute. This is in the same range as Kraken in normal operation, which can process 1.3 million 100 bp reads per minute. Additionally, the prototype managed to identify all of the species that were present in the simulated sample. A small fraction of the reads, however, was misclassified as organisms that are very abundant in UniProt, but of which no close relatives were present in the original sample.

Further investigation showed that many of the short peptides were responsible for the misclassified reads by accidentally mapping to sequences in UniProt. In an effort to filter these erroneous peptides using machine learning, a random forest was trained using peptide length and amino acid composition as main features.

**Table 6.1** Comparison of the performance of the classifiers as tested by Wood and Salzberg (2014) with the prototype of the Unipept metagenomics pipeline (with and without filtering). The filtering, as implemented by the random forest, increases the precision dramatically without reducing sensitivity. The speed and precision of Unipept is comparable to Kraken, but the sensitivity is many times lower.

| CLASSIFIER | PRECISION | SENSITIVITY | SPEED (READS/MIN) |
|---|---|---|---|
| Naïve Bayes Classifier | 97.64% | 97.64% | 7 |
| PhymmBL | 96.11% | 96.11% | 76 |
| PhymmBL (conf. > 0.65) | 99.08% | 95.45% | 76 |
| Megablast w/ best hit | 96.93% | 93.67% | 4 511 |
| Kraken | 99.90% | 91.25% | 1 307 161 |
| MiniKraken (Kraken w/ 4GB DB) | 99.95% | 65.87% | 1 441 476 |
| MetaPhlAn | n/a | n/a | 370 770 |
| **Unipept** | 73.69% | 17.66% | ~1 200 000 |
| **Unipept with filtering** | 96.10% | 17.38% | ~1 200 000 |

To test the effectiveness of the filter, we ran the benchmark that was used in Wood and Salzberg (2014) to compare Kraken with other tools. As can be seen in Table 6.1, the filter managed to increase the precision dramatically from 73.69% to 96.10% without further reducing the (admittedly low) sensitivity. The overall speed of the Unipept pipeline is already comparable to Kraken and could even be improved by using a local index instead of relying on the remote web server.

The low sensitivity has a few diverse reasons. The second step in the pipeline is predicting proteins using FragGeneScan. Although FragGeneScan is one of the best tools for the job, only 85% of the input reads gets a prediction. This low prediction level results in an immediate 15% loss in sensitivity. There are two main reasons for this: not all DNA is coding and read errors may prevent a reliable translation. This is a disadvantage of working in the protein space for which there is no easy solution. A second source of sensitivity loss is the depth of identification. The result table (Table 6.1) only looks at the sensitivity for identifications made at the genus level or better, which is only about half of the reads that are identified by Unipept. Of the other half that was identified at a less specific taxonomic level, only 0.07% was incorrect. A third reason are the read errors in the source data. The reads in the benchmark are 100 bp reads with high sequencing error (2.1% SNP rate, 1.1% indel rate). Because Unipept uses exact matching, a single error can have the effect that Unipept can't identify a single peptide in a read and therefore can't identify the read at all. Switching from tryptic peptides to k-mers will probably have a positive effect on the second and third case and improve sensitivity.

### Conclusion

The Unipept toolbox can thus be expanded with a metagenomics pipeline for the taxonomic identification of DNA reads from shotgun metagenomics data sets (at least with the existing and novel strategies discussed above), expanding the scope of the platform from metaproteomics to metagenomics. For this, a benchmark study can be performed to compare the speed of execution, storage requirements and accuracy of the different strategies (including the traditional identification approaches implemented in other metagenomics tools), using real and simulated shotgun metagenome data. A reasonable target seems that Unipept should be able to accurately identify at least 1 million 500bp DNA reads per minute, where other packages take hours to perform the same computations. This would enable us, in collaboration with the EBI Metagenomics team, to re-analyze all metagenomics samples from the EBI Metagenomics Archive at regular time interval (e.g., monthly). At present, each data set submitted to this archive is analyzed only once at the time of submission, where reference databases grow at an exponential rate leading to less under-sampled biodiversity.

Current NGS technologies allow metagenomes to be sequenced much deeper than metaproteomes, which can provide a more detailed insight into the complex biodiversity of the samples. Combined metagenomes and metaproteomes of the same sample will also allow to link the functional potential encoded in genomes to expression levels measured in the sample, and to associate differential functional expressions with shifts in the biodiversity. Moreover, to obtain optimal identification of the peptides resulting from metaproteomics experiments, most studies currently match peptides against shotgun metagenomics data sets from the same samples (Erickson et al., 2012; Kolmeder and Vos, 2014).

## 6.2 Functional analysis of metaproteomics data

At the heart of Unipept lies an index structure for fast mapping of tryptic peptides onto all UniProt protein sequences having exact substring matches (Figure 3.4). The taxonomic and functional annotations on matched UniProt entries can be used to infer taxonomic and functional assignments for the individual peptides from metaproteomics experiments using the Tryptic Peptide Analysis feature of Unipept (http://unipept.ugent.be/search/single). The Shotgun Metaproteomics Analysis Pipeline (http://unipept.ugent.be/datasets) of Unipept currently supports streamlined identification, analysis, and visualization of all peptides from a metaproteomics experiment, and as described in Section 6.1 this pipeline can be extended for biodiversity analysis of shotgun metagenomics experiments. This diversity analysis could be extended with a statistical data analysis and visualization framework for functional analysis of metagenomics and metaproteomics experiments.

### Aggregation strategies

The traditional approach for functional analysis of shotgun metagenomics/metaproteomics data sets provided by current tools is to represent functional ontology annotations (e.g., Gene Ontology, GO; Ashburner et al. (2000)) as charts that are often zoomable per ontological level, or to map Enzyme Commission (EC; Bairoch (2000)) numbers onto metabolic pathways such as those provided by the Kyoto Encyclopedia of Genes and Genomes (KEGG; Kanehisa and Goto (2000)).

The implementation of this kind of functional and metabolic pathway analysis tools in Unipept will be quite straightforward, as fast peptide-to-protein matching is already part of the Unipept kernel, functional assignments of all UniProt entries are already parsed in the Unipept database, and Unipept already has implemented flexible and interactive visualizations that can be reused for basic functional analysis. However, although many metagenomics and metaproteomics papers base their discussion of the functional complement of environmental samples on GO pie charts and/or metabolic pathway expression levels, we dare to claim that such visualizations usually do not provide deep biological insight. We therefore see an implementation of the basis functional analysis tools only as a stepping stone for the functional analysis framework of Unipept, that can both be made more accurate by taking advantage of aggregation strategies as outlined in Section 6.1 and can lead to in-depth comparative functional analysis (discussed in Section 6.3).

The aggregation strategies for taxonomic assignments as discussed in Section 6.1 cannot directly be applied for the aggregation of multiple GO assignments (e.g., taken from all proteins matching a peptide sequence, or taken from all individual peptides in a (partial) protein predicted on a DNA read), as GO is structured as a directed acyclic graph (DAG) rather than a tree (hierarchy). Additionally, a single reference protein may have multiple GO annotations in contrast to a single taxonomic assignment. However, the weighted hierarchical distribution algorithm discussed in Section 6.1 (of which the LCA and maximal RTL algorithms are special cases) can be extended into a weighted DAG distribution algorithm (Figure 6.3). In this case, top-down redistribution of the weight of a parent node (step 2 in the hierarchical algorithm) might have multiple contributions to the same leaf node as a consequence of multiple parallel node-to-leaf paths. Bottom-up progression of leaf weights (step 4 in the hierarchical algorithm) might have to split a child weight over multiple parent nodes.

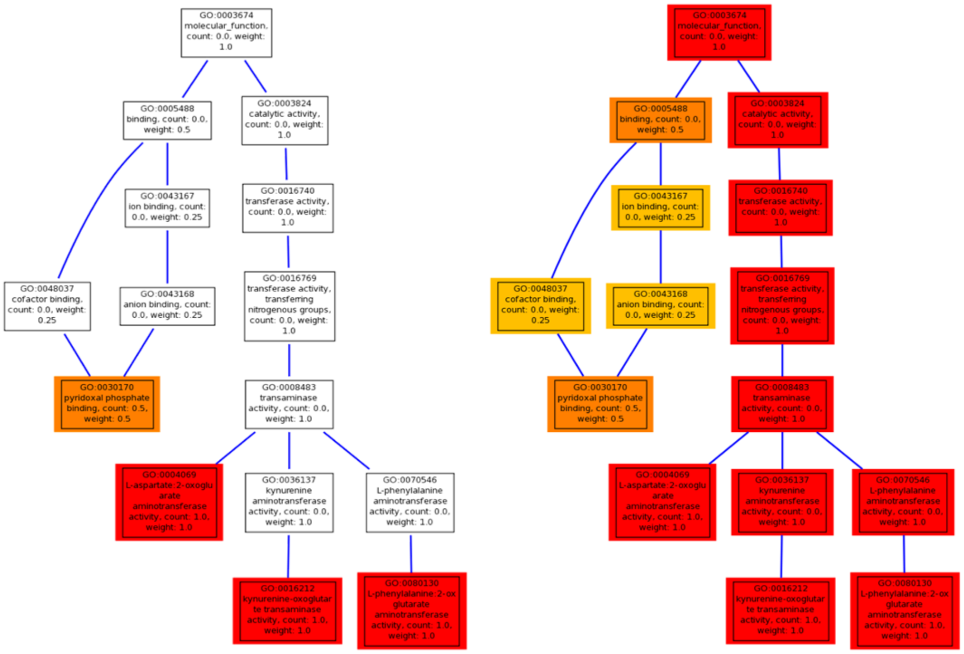

GO:0003674 molecular_function, count: 0.0, weight: 1.0
GO:0005488 binding, count: 0.0, weight: 0.5
GO:0003824 catalytic activity, count: 0.0, weight: 1.0
GO:0043167 ion binding, count: 0.0, weight: 0.25
GO:0016740 transferase activity, count: 0.0, weight: 1.0
GO:0048037 cofactor binding, count: 0.0, weight: 0.25
GO:0043168 anion binding, count: 0.0, weight: 0.25
GO:0016769 transferase activity, transferring nitrogenous groups, count: 0.0, weight: 1.0
GO:0030170 pyridoxal phosphate binding, count: 0.5, weight: 0.5
GO:0008483 transaminase activity, count: 0.0, weight: 1.0
GO:0004069 L-aspartate:2-oxoglutarate aminotransferase activity, count: 1.0, weight: 1.0
GO:0036137 kynurenine aminotransferase activity, count: 0.0, weight: 1.0
GO:0070546 L-phenylalanine aminotransferase activity, count: 0.0, weight: 1.0
GO:0016212 kynurenine-oxoglutarate transaminase activity, count: 1.0, weight: 1.0
GO:0080130 L-phenylalanine:2-oxoglutarate aminotransferase activity, count: 1.0, weight: 1.0
GO:0003674 molecular_function, count: 0.0, weight: 1.0
GO:0005488 binding, count: 0.0, weight: 0.5
GO:0003824 catalytic activity, count: 0.0, weight: 1.0
GO:0043167 ion binding, count: 0.0, weight: 0.25
GO:0016740 transferase activity, count: 0.0, weight: 1.0
GO:0048037 cofactor binding, count: 0.0, weight: 0.25
GO:0043168 anion binding, count: 0.0, weight: 0.25
GO:0016769 transferase activity, transferring nitrogenous groups, count: 0.0, weight: 1.0
GO:0030170 pyridoxal phosphate binding, count: 0.5, weight: 0.5
GO:0008483 transaminase activity, count: 0.0, weight: 1.0
GO:0004069 L-aspartate:2-oxoglutarate aminotransferase activity, count: 1.0, weight: 1.0
GO:0036137 kynurenine aminotransferase activity, count: 0.0, weight: 1.0
GO:0070546 L-phenylalanine aminotransferase activity, count: 0.0, weight: 1.0
GO:0016212 kynurenine-oxoglutarate transaminase activity, count: 1.0, weight: 1.0
GO:0080130 L-phenylalanine:2-oxoglutarate aminotransferase activity, count: 1.0, weight: 1.0

**Figure 6.3** Example of applying the weighted DAG distribution algorithm, showing the initial frequencies of GO term annotations of all UniProt proteins that match the tryptic peptides SSWWAHVEMGPPDPIL-GVTEAYK or DTNSK (left) and the result after redistributing the weights top-down and bottom-up (right).

This is a generalization of the aggregation strategy implemented in the commercial software package Blast2GO (Conesa et al., 2005), from which we can use the idea to incorporate Evidence Code weights to promote the assignment of functional annotations with experimental evidence and penalize electronic annotations or low traceability. The latter is important, since it has been predicted that 13-15% of the functional annotations in reference databases contain database propagation errors (Brenner, 1999).

Similar to the algorithm described in Section 6.1, we can avoid using Blast by relying on the Unipept index. As with the LCAs of taxonomic assignments, aggregation of functional assignments of the proteins matching a tryptic peptide can be precomputed for all peptides extracted from

UniProt and cached in the Unipept database to increase performance of downstream analysis. One of the challenges will be to come up with fast implementations of functional aggregation strategies for metaproteomics and metagenomics data sets, and to benchmark the performance and accuracy of the different alternatives along the lines of the taxonomic benchmark outlined in Section 6.1.

**Applications**

This Unipept extension can be applied for the functional and metabolic pathway analysis of the metagenomics and metagenomics data sets, for example data sets generated from the faecal samples of CF patients and their healthy siblings (Debyser et al., 2016). The most pronounced observation in diseases associated with gut dysbiosis (inflammatory bowel disease, colon cancer) is a substantial decrease of "health promoting" species such as *Faecalibacterium prausnitzii*. This was also observed in our cross-sectional shotgun metaproteomics analysis of faecal samples from CF patients (Debyser et al., 2016).

These organisms are typical producers of butyrate, a key metabolite used as energy source by colonocytes that has anti-inflammatory properties. It therefore seems like a logical conclusion that dysbiosis is associated with a lack of butyrate production. However, systematic analysis has shown recently that more bacteria have butyrate-producing pathways than was previously anticipated, and that these organisms show high abundance (up to 40%) in stool samples of healthy individuals investigated by the HMP consortium (Vital, Howe, and Tiedje, 2014). Causal inferences between depletion of certain taxa and observed functional shifts should therefore be drawn carefully.

As metaproteomics provides direct abundance measurements of the enzymes involved in metabolic pathways, this gives a better reflection of the potential changes in biochemical pathways. The functional extensions of Unipept could therefore be applied for the analysis of shotgun metaproteomics data sets that were previously generated from the faecal samples of CF patients and their unaffected siblings. This then could answer the question whether dysbiosis indeed results in a loss of enzymes associated with butyrate production. Other interesting questions regarding functional changes due to dysbiosis in CF patients include a potential raise in expression of antibiotic resistance genes, mucin degrading extracellular glycosidases, and increased abundance of proteins involved in inflammatory pathways at level of host proteins.

## 6.3 Comparative analysis of metaproteomics data

Recent review papers about the current state-of-the-art in metaproteomics praise Unipept for the performance and accuracy of its taxonomic identification pipeline, and also for its interactive visualization framework that helps to explore the biodiversity in complex environmental samples (Seifert et al., 2013; Kolmeder and Vos, 2014). However, most environmental studies do not merely apply metagenomics and metaproteomics to gain insight in the taxonomic, functional, and metabolic composition of individual samples, but rather want to investigate observed compositional shifts between multiple samples.

In general, there are two types of environmental studies. Cross-sectional studies investigate samples collected from distinct environmental niches at one specific point in time or under a time-independence assumption in order to observe causal effects of one or more environmental factors upon sample composition. Longitudinal studies involve repeated observations of the same environmental niche over a period of time to study shifts in sample composition that may be correlated to certain temporal events.

**A comparative analysis framework**

Unipept could be expanded to include a framework for statistical analysis and data visualization of multiple metagenomics or metaproteomics data sets from cross-sectional and longitudinal studies (Mehlan et al., 2013; Wang et al., 2015). As showcased by Unipept, such a statistical analysis and data visualization toolbox nowadays can be implemented as a highly responsive client application hosted in a web browser using the latest web technologies. In particular, when making use of the latest HTML5 technologies including Web Workers for parallel processing and Local Storage for browser caching, as well as JavaScript libraries including D3.js for interactive data manipulation and visualization.

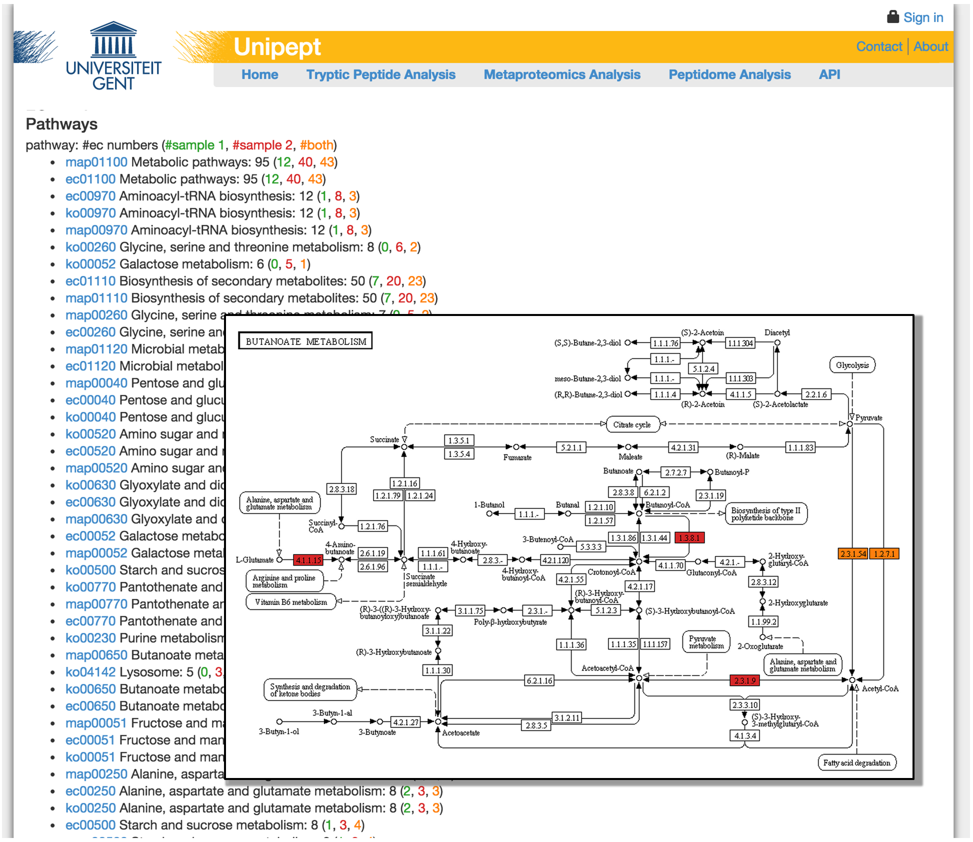
This framework could allow third-party developers to reuse individual components in their own applications. This will allow the statistical analysis and data visualization framework to be used as a standalone client application that enables users to upload multi-sample contingency tables that were processed by metagenomics or metaproteomics pipelines (e.g., in the Biological Observation Matrix (BIOM) format (McDonald et al. (2012)) which is an interoperable format supported by all major pipelines) or to embed the framework in other web applications.

The framework could support basic visualizations such as pie charts, (stacked) bar charts and heat maps, but also more elaborate statistical and visualization approaches for comparative analysis such as Lefse (Segata et al., 2011) for differential taxonomic and functional analysis and UniFrac (Lozupone and Knight, 2005) to calculate distances between organismal communities using phylogenetic information.

In contrast to the existing MetaSee framework (Song et al., 2012), such framework should be implemented according to the Model-View-Controller (MVC) design pattern. This pattern allows to store the data only once (model), with the possibility of adding interactive operations (controllers) to navigate the data visualizations (views). As such, additional components can be plugged into the framework as independent controllers or viewers. This will require the definition of a standard data representation layer (tabular, tree or (directed acyclic) graph data structures) and a standard data manipulation interface. For example, many metagenomics tools visualize the biodiversity in a sample not as a tree (which may include simply too much information), but sliced at a specific taxonomic rank (e.g., as a pie chart at the phylum level). From the MVC perspective, this slice operation is not implemented as a static conversion from a tree format to a tabular format but as a dynamic view of the tree as a table, which allows more dynamic visualizations that navigate top-down and bottom-up in the tree. Moreover, the same controller can be applied to all sample data sets at once, allowing interactive comparative visualizations.

## Prototype

An initial prototype of a comparative functional analysis tool has already been implemented and can be seen in Figure 6.4. In its current implementation, aggregated functional assignments of the tryptic peptides of one or two samples are mapped onto KEGG pathways using the EC numbers that are cross-referenced in UniProt. Here, the pathways are heuristically ranked on interestingness, based on the number of differentially expressed enzymatic reactions.



## Pathways

pathway: #ec numbers (#sample 1, #sample 2, #both)

- map01100 Metabolic pathways: 95 (12, 40, 43)
- ec01100 Metabolic pathways: 95 (12, 40, 43)
- ec00970 Aminoacyl-tRNA biosynthesis: 12 (1, 8, 3)
- ko00970 Aminoacyl-tRNA biosynthesis: 12 (1, 8, 3)
- map00970 Aminoacyl-tRNA biosynthesis: 12 (1, 8, 3)
- ko00260 Glycine, serine and threonine metabolism: 8 (0, 6, 2)
- ko00052 Galactose metabolism: 6 (0, 5, 1)
- ec01110 Biosynthesis of secondary metabolites: 50 (7, 20, 23)
- map01110 Biosynthesis of secondary metabolites: 50 (7, 20, 23)
- map00260 Glycine, serine
- ec00260 Glycine, serine an
- map01120 Microbial metab
- ec01120 Microbial metabol
- map00040 Pentose and glu
- ec00040 Pentose and glucu
- ko00040 Pentose and glucu
- ko00520 Amino sugar and
- ec00520 Amino sugar and
- map00520 Amino sugar an
- ko00630 Glyoxylate and di
- ec00630 Glyoxylate and di
- map00630 Glyoxylate and
- ec00052 Galactose metab
- map00052 Galactose meta
- ko00500 Starch and sucros
- ko00770 Pantothenate and
- map00770 Pantothenate a
- ec00770 Pantothenate and
- ko00230 Purine metabolism
- map00650 Butanoate meta
- ko04142 Lysosome: 5 (0, 3
- ko00650 Butanoate metab
- ec00650 Butanoate metab
- map00051 Fructose and m
- ec00051 Fructose and man
- ko00051 Fructose and man
- map00250 Alanine, asparta
- ec00250 Alanine, aspartate and glutamate metabolism: 8 (2, 3, 3)
- ko00250 Alanine, aspartate and glutamate metabolism: 8 (2, 3, 3)
- ec00500 Starch and sucrose metabolism: 8 (1, 3, 4)

**Figure 6.4** Prototype for comparative functional analysis in Unipept using EC numbers mapped on KEGG pathways.

The prototype requires a number of extension before it can be added to an official Unipept release. First of all, as discussed in Section 6.2, the functional assignments could equally well be based on the DNA reads from metagenomics samples. The prototype can also be extended to support cross-sectional studies that want to compare the functional shifts between multiple samples from two distinct environmental niches.

Instead of the current ranking algorithm, a more statistically sound method should be implemented. Examples are the statistical framework for differentially expressed proteins as implemented in MetaStats (White, Nagarajan, and Pop, 2009) and its extension MetaPath (Liu and Pop, 2011), which allows much more granular detection of subpathways that are globally over or under expressed between subpopulations. The current third-party implementation of these tools is quite slow, but it seems that a fast reimplementation might be achievable.

Another goal could be to allow integrated taxonomic and functional analysis. Current metagenomics and metaproteomics tools implement taxonomic and functional analysis merely as separate pipelines, whereas biological comparison between the samples would greatly benefit from statistical tools and visualization aids that intertwine both viewpoints. As taxonomic and functional assignments are made for the individual peptides (metaproteomics) or DNA reads (metagenomics), they establish a complex graph (network) where both pieces of the puzzle come together. However, tools for visualizing metabolic pathways as KEGG or iPath (Yamada et al., 2011) do not provide additional annotations to highlight “who is doing what” in the environment. Having integrated visualizations and graph-based statistical tools would allow to search for functional changes beyond what can be explained by a mere shift in populations, and vice versa, to search for shifts in populations that do not result in functional changes.

## 6.4 Targeted metaproteomics

The proteomics field is recently moving into more targeted approaches to address specific biological questions. Typically, this involves Single or Multiple Reaction Monitoring

(SRM/MRM; Pan et al. (2012)) employing triple quadrupole-like instruments to monitor protein specific tryptic fragments, focusing on a combined set of the precursor ion masses and their MS/MS derived fragments (transitions). Target peptides can be selected according to a wide range of criteria such as their protein-specificity to screen for the presence/absence of certain proteins in proteomics studies. Absolute quantification is achieved by providing complementary isotope labeled peptide standards and the approach is increasingly used in clinical biomarker analysis.

A growing number of human or animal diseases are associated with perturbation of the microbiota naturally residing in the skin or gut of the host. The presence/absence of representatives of specific bacterial genera affects the host condition, and therefore combining host and microbiota biomarkers could be useful in diagnostics in diseases such as Inflammatory Bowel Disease, diabetes or cystic fibrosis, and in microbial contamination detection in food safety. However, selection of suitable peptide markers for these bacteria requires taxon-specificity. With protein profiling providing assays closer to activated functions, metaproteome-wide association studies have the potential to become an important tool in environmental, clinical, and food studies (Juste et al., 2014).

The Unique Peptide Finder (http://unipept.ugent.be/peptidefinder; Mesuere et al. (2016a)) of Unipept already allows to determine the unique tryptic peptides (the so-called unique peptidomes) of a certain species or pool of species (represented by a selection of available whole-genome sequences). These are the peptides present in all of the selected genome sequences but are found nowhere else outside the species of the selected genomes (actually the LCA of all taxonomic annotations of the selected genomes). These peptides are therefore ideal biomarkers. However, the Unique Peptide Finder typically yields over a thousand unique peptides for a species and not all of them are equally suitable for MRM analysis. Parameters such as, peptide length, presence of amino acids that are prone to chemical modification, and even the cost to produce synthetic peptides for method optimization or stable isotope labeled peptides as internal standards are additional criteria that need to be taken into account. Because the Unique Peptide Finder returns hundreds of potentially useful peptides, we can afford to use strict filters to only retain the most suitable peptide targets.

Several computational tools have however, been built to help select the most suitable peptides. These tools each consider a specific property of the peptide in question: informativeness, presence, ionization, and fragmentation. Informativeness is crucial, because a unique sequence is not a sufficient condition to ensure unambiguous assignment of the detected mass spectrometry signal to a given peptide, and thence a protein and a species. For this, the recorded signal itself must be unique. Two existing tools can already compute these peptide properties: SigPep (Helsens et al., 2012) and SRMCollider (Rost, Malmstrom, and Aebersold, 2012). SigPep is currently undergoing an extension to take advantage of accurate retention time prediction from ELUDE (Moruz et al., 2012) as well. This can be further optimized using optimal gradient choices (Bertsch et al., 2010). Presence of a peptide in the sample after digestion is of course affected by presence of the parent protein, but is also strongly affected by the propensity of trypsin (the leading protease in proteomics; Vandermarliere, Mueller, and Martens (2013)) to cleave this peptide from the protein. Here too, predictors exist, as reviewed in (Kelchtermans et al., 2014), with the optimal predictor so far being CP-DT (Fannes et al., 2013). Interestingly, CP-DT also provides a good estimate of the probability of ionization for a given peptide. Finally, the fragmentation pattern that will result from a given peptide can now also be predicted with high accuracy, both for CID as well as HCD fragmentation (Degroeve, Maddelein, and Martens, 2015).

The addition of these predictions will help reduce the large number of possible MRM peptide targets obtained from the Unique Peptide Finder, by allowing the user to immediately select the most promising set of peptides. As a result, users will be able to move from query to actual MRM assay much more efficiently.

# References

Acland, Abigail, Richa Agarwala, Tanya Barrett, Jeff Beck, Dennis a. Benson, Colleen Bollin, Evan Bolton, et al. 2014. "Database resources of the National Center for Biotechnology Information." *Nucleic Acids Research* 42 (D1): 7–17. doi:10.1093/nar/gkt1146.

Altschul, S F, W Gish, W Miller, E W Myers, and D J Lipman. 1990. "Basic local alignment search tool." *Journal of Molecular Biology* 215 (3): 403–10. doi:10.1016/S0022-2836(05)80360-2.

Andrews, Keith, and Helmut Heidegger. 1998. "Information Slices: Visualising and Exploring Large Hierarchies using Cascading, Semi-Circular Discs." In *IEEE Symposium on Information Visualization. InfoVis.*, 4–7.

Ashburner, M, C A Ball, J A Blake, D Botstein, H Butler, J M Cherry, A P Davis, et al. 2000. "Gene ontology: tool for the unification of biology. The Gene Ontology Consortium." *Nature Genetics* 25 (1): 25–29. doi:10.1038/75556.

Askenazi, Manor, Jarrod A. Marto, and Michal Linial. 2010. "The complete peptide dictionary - A meta-proteomics resource." *Proteomics* 10 (23): 4306–10. doi:10.1002/pmic.201000270.

Bairoch, A. 2000. "The ENZYME database in 2000." *Nucleic Acids Research* 28 (1): 304–5. doi:10.1093/nar/28.1.304.

Bairoch, Amos, Rolf Apweiler, Cathy H. Wu, Winona C. Barker, Brigitte Boeckmann, Serenella Ferro, Elisabeth Gasteiger, Hongzhan Huang, Rodrigo Lopez, Michele Magrane, Maria J. Martin, Darren A. Natale, Claire O'Donovan, Nicole Redaschi, and Lai Su L Yeh. 2005. "The Universal Protein Resource (UniProt)." *Nucleic Acids Research* 33: D154–D159. doi:10.1093/nar/gki070.

Balme, D.M. 1962. "Development of Biology in Aristotle and Theophrastus: Theory of Spontaneous Generation." *Phronesis: A Journal for Ancient Philosophy* 7 (1): 91–104. doi:10.1163/156852862X00052.

Battistuzzi, Fabia U, Andreia Feijao, and S Blair Hedges. 2004. "A genomic timescale of prokaryote evolution: insights into the origin of methanogenesis, phototrophy, and the colonization of land." *BMC Evolutionary Biology* 4 (1): 44. doi:10.1186/1471-2148-4-44.

Bazinet, Adam L, and Michael P Cummings. 2012. "A comparative evaluation of sequence classification programs." *BMC Bioinformatics* 13 (1): 92. doi:10.1186/1471-2105-13-92.

Benson, Dennis a., Mark Cavanaugh, Karen Clark, Ilene Karsch-Mizrachi, David J. Lipman, James Ostell, and Eric W. Sayers. 2013. "GenBank." *Nucleic Acids Research* 41 (D1): 36–42. doi:10.1093/nar/gks1195.

Bernstein, Harris, and Carol Bernstein. 2012. *DNA repair as the primary adaptive function of sex in bacteria and eukaryotes*.

Bertsch, Andreas, Stephan Jung, Alexandra Zerck, Nico Pfeifer, Sven Nahnsen, Carsten Henneges, Alfred Nordheim, and Oliver Kohlbacher. 2010. "Optimal de novo design of MRM experiments for rapid assay development in targeted proteomics." *Journal of Proteome Research* 9 (5): 2696–2704. doi:10.1021/pr1001803.

Boeckmann, Brigitte, Amos Bairoch, Rolf Apweiler, Marie Claude Blatter, Anne Estreicher, Elisabeth Gasteiger, Maria J. Martin, Karine Michoud, Claire O'Donovan, Isabelle Phan, Sandrine Pilbout, and Michel Schneider. 2003. "The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003." *Nucleic Acids Research* 31 (1): 365–70. doi:10.1093/nar/gkg095.

Bostock, Michael, Vadim Ogievetsky, and Jeffrey Heer. 2011. "D3 Data-Driven Documents." *IEEE Transactions on Visualization and Computer Graphics* 17 (12): 2301–9. doi:10.1109/TVCG.2011.185.

Bremer, B., K. Bremer, M. W. Chase, M. F. Fay, J. L. Reveal, L. H. Bailey, D. E. Soltis, P. S. Soltis, and P. F. Stevens. 2009. "An update of the Angiosperm Phylogeny Group classification for the orders and families of flowering plants: APG III." *Botanical Journal of the Linnean Society* 161 (2): 105–21. doi:10.1111/j.1095-8339.2009.00996.x.

Brenner, Steven E. 1999. "Errors in genome annotation." *Trends in Genetics* 15 (4): 132–33. doi:10.1016/S0168-9525(99)01706-0.

Butterfield, Nicholas J. 2000. "Bangiomorpha pubescens n. gen., n. sp.: implications for the evolution of sex, multicellularity, and the Mesoproterozoic/Neoproterozoic radiation of eukaryotes." *Paleobiology* 26 (3): 386–404. doi:10.1666/0094-8373(2000)026<0386:BPNGNS>2.0.CO;2.

Chen, Chuming, Zhiwen Li, Hongzhan Huang, Baris E. Suzek, and Cathy H. Wu. 2013. "A fast peptide match service for UniProt knowledgebase." *Bioinformatics* 29 (21): 2808–9. doi:10.1093/bioinformatics/btt484.

Chenau, Jérôme, François Fenaille, Valérie Caro, Michel Haustant, Laure Diancourt, Silke R Klee, Christophe Junot, Eric Ezan, Pierre L Goossens, and François Becher. 2014. "Identification and validation of specific markers of Bacillus anthracis spores by proteomics and genomics approaches." *Molecular & Cellular Proteomics : MCP* 13 (3): 716–32. doi:10.1074/mcp.M113.032946.

Conesa, Ana, Stefan Götz, Juan Miguel García-Gómez, Javier Terol, Manuel Talón, and Montserrat Robles. 2005. "Blast2GO: A universal tool for annotation, visualization and analysis in functional genomics research." *Bioinformatics* 21 (18): 3674–6. doi:10.1093/bioinformatics/bti610.

Craig, Robertson, and Ronald C Beavis. 2003. "A method for reducing the time required to match protein sequences with tandem mass spectra." *Rapid Commun Mass Spectrom* 17 (20): 2310–6. doi:10.1002/rcm.1198.

Darwin, Charles. 1859. *On the Origin of Species*. Vol. 5. doi:10.1016/S0262-4079(09)60380-8.

Debyser, Griet, Bart Mesuere, Lieven Clement, Jens Van de Weygaert, Pieter Van Hecke, Gwen Duytschaever, Maarten Aerts, Peter Dawyndt, Kris De Boeck, Peter Vandamme, and Bart Devreese. 2016. "Faecal proteomics: A tool to investigate dysbiosis and inflammation in patients with cystic fibrosis." *Journal of Cystic Fibrosis : Official Journal of the European Cystic Fibrosis Society* 15 (2). Elsevier: 242–50. doi:10.1016/j.jcf.2015.08.003.

Degroeve, Sven, Davy Maddelein, and Lennart Martens. 2015. "MS <sup>2</sup> PIP prediction server: compute and visualize MS <sup>2</sup> peak intensity predictions for CID and HCD fragmentation." *Nucleic Acids Research* 43 (W1): W326–W330. doi:10.1093/nar/gkv542.

Delmotte, Nathanaël, Claudia Knief, Samuel Chaffron, Gerd Innerebner, Bernd Roschitzki, Ralph Schlapbach, Christian von Mering, and Julia A Vorholt. 2009. "Community proteogenomics reveals insights into the physiology of phyllosphere bacteria." *Proceedings of the National Academy of Sciences of the United States of America* 106 (38): 16428–33. doi:10.1073/pnas.0905240106.

Dicksved, Johan, Jonas Halfvarson, Magnus Rosenquist, Gunnar Järnerot, Curt Tysk, Juha Apajalahti, Lars Engstrand, and Janet K Jansson. 2008. "Molecular analysis of the gut microbiota of identical twins with Crohn's disease." *The ISME Journal* 2 (7): 716–27. doi:10.1038/ismej.2008.37.

Edgar, Robert C. 2010. "Search and clustering orders of magnitude faster than BLAST." *Bioinformatics* 26 (19): 2460–1. doi:10.1093/bioinformatics/btq461.

Eidhammer, Ingvar, Kristian Flikka, Lennart Martens, and Svein-Ole Mikalsen. 2007. "Protein, proteome, and proteomics." In *Computational Methods for Mass Spectrometry Proteomics*, 1–29.

Elgar, Greg, and Tanya Vavouri. 2008. "Tuning in to the signals: noncoding sequence conservation in vertebrate genomes." *Trends in Genetics* 24 (7): 344–52. doi:10.1016/j.tig.2008.04.005.

Eng, J K, A L McCormack, and J R Yates. 1994. "An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database." *Journal of the American Society for Mass Spectrometry* 5 (11): 976–89. doi:10.1016/1044-0305(94)80016-2.

Erickson, Alison R, Brandi L Cantarel, Regina Lamendella, Youssef Darzi, Emmanuel F Mongodin, Chongle Pan, Manesh Shah, Jonas Halfvarson, Curt Tysk, Bernard Henrissat, Jeroen Raes, Nathan C Verberkmoes, Claire M Fraser, Robert L Hettich, and Janet K Jansson. 2012. "Integrated metagenomics/metaproteomics reveals human host-microbiota signatures of Crohn's disease." *PloS One* 7 (11): e49138. doi:10.1371/journal.pone.0049138.

Euzéby, J P. 1997. "List of Bacterial Names with Standing in Nomenclature: a folder available on the Internet." *International Journal of Systematic Bacteriology* 47 (2): 590–92. doi:10.1099/00207713-47-2-590.

Fannes, Thomas, Elien Vandermarliere, Leander Schietgat, Sven Degroeve, Lennart Martens, and Jan Ramon. 2013. "Predicting tryptic cleavage from proteomics data using decision tree ensembles." *Journal of Proteome Research* 12 (5): 2253–9. doi:10.1021/pr4001114.

Federhen, Scott. 2012. "The NCBI Taxonomy database." *Nucleic Acids Research* 40: D136–43. doi:10.1093/nar/gkr1178.

Federhen, Scott, Karen Clark, Tanya Barrett, Helen Parkinson, James Ostell, Yuichi Kodama, Jun Mashima, Yasukazu Nakamura, Guy Cochrane, and Ilene Karsch-Mizrachi. 2014. "Toward richer metadata for microbial sequences: replacing strain-level NCBI taxonomy taxids with BioProject, BioSample and Assembly records." *Standards in Genomic Sciences* 9 (3): 1275–7. doi:10.4056/sigs.4851102.

Fenn, J B, M Mann, C K Meng, S F Wong, and C M Whitehouse. 1989. "Electrospray ionization for mass spectrometry of large biomolecules." *Science* 246 (4926): 64–71. doi:10.1126/science.2675315.

Fiers, W, R Contreras, F Duerinck, G Haegeman, D Iserentant, J Merregaert, W Min Jou, F Molemans, a Raeymaekers, a Van den Berghe, G Volckaert, and M Ysebaert. 1976. "Complete nucleotide sequence of bacteriophage MS2 RNA: primary and secondary structure of the replicase gene." *Nature* 260 (5551): 500–507. doi:10.1038/260500a0.

Flannery, D. T., and M. R. Walter. 2012. "Archean tufted microbial mats and the Great Oxidation Event: new insights into an ancient problem." *Australian Journal of Earth Sciences* 59 (1): 1–11. doi:10.1080/08120099.2011.607849.

Fournier, Pierre-Edouard, David Vallenet, Valérie Barbe, Stéphane Audic, Hiroyuki Ogata, Laurent Poirel, Hervé Richet, Catherine Robert, Sophie Mangenot, Chantal Abergel, Patrice Nordmann, Jean Weissenbach, Didier Raoult, and Jean-Michel Claverie. 2006. "Comparative genomics of multidrug resistance in Acinetobacter baumannii." *PLoS Genetics* 2 (1): e7. doi:10.1371/journal.pgen.0020007.

Garwood, Russell J. 2012. "Patterns In Palaeontology: The first 3 billion years of evolution." http://www.palaeontologyonline.com/articles/2012/patterns-in-palaeontology-the-first-3-billion-years-of-evolution/.

Geer, Lewis Y., Sanford P. Markey, Jeffrey A. Kowalak, Lukas Wagner, Ming Xu, Dawn M. Maynard, Xiaoyu Yang, Wenyao Shi, and Stephen H. Bryant. 2004. "Open mass spectrometry search algorithm." *Journal of Proteome Research* 3 (5): 958–64. doi:10.1021/pr0499491.

Gest, Howard. 2004. "The discovery of microorganisms by Robert Hooke and Antoni Van Leeuwenhoek, fellows of the Royal Society." *Notes and Records of the Royal Society of London* 58 (2): 187–201. doi:10.1098/rsnr.2004.0055.

Goodman, M, C a Porter, J Czelusniak, S L Page, H Schneider, J Shoshani, G Gunnell, and C P Groves. 1998. "Toward a phylogenetic classification of Primates based on DNA evidence complemented by fossil evidence." *Molecular Phylogenetics and Evolution* 9 (3): 585–98. doi:10.1006/mpev.1998.0495.

Hammond, Peter. 1992. "Species inventory." In *Global Biodiversity: Status of the Earth's Living Resources*, 17–39.

Helsens, Kenny, Michael Mueller, Niels Hulstaert, and Lennart Martens. 2012. "Sigpep: Calculating unique peptide signature transition sets in a complete proteome background." *Proteomics* 12 (8): 1142–6. doi:10.1002/pmic.201100566.

Herbst, Florian-Alexander, Vanessa Lünsmann, Henrik Kjeldal, Nico Jehmlich, Andreas Tholey, Martin von Bergen, Jeppe Lund Nielsen, Robert L. Hettich, Jana Seifert, and Per Halkjaer Nielsen. 2016. "Enhancing Metaproteomics - The value of models and defined environmental microbial systems." *Proteomics* 16 (5): 783–98. doi:10.1002/pmic.201500305.

Hettich, Robert L., Chongle Pan, Karuna Chourey, and Richard J. Giannone. 2013. "Metaproteomics: Harnessing the power of high performance mass spectrometry to identify the suite of proteins that control metabolic activities in microbial communities." *Analytical Chemistry* 85 (9): 4203–14. doi:10.1021/ac303053e.

Hirosawa, M, M Hoshida, M Ishikawa, and T Toya. 1993. "MASCOT: multiple alignment system for protein sequences based on three-way dynamic programming." *Computer Applications in the Biosciences : CABIOS* 9 (2): 161–67. doi:10.1093/bioinformatics/9.2.161.

Hu, Qizhi, Robert J. Noll, Hongyan Li, Alexander Makarov, Mark Hardman, and R. Graham Cooks. 2005. "The Orbitrap: A new mass spectrometer." *Journal of Mass Spectrometry* 40 (4): 430–43. doi:10.1002/jms.856.

Hunter, Sarah, Matthew Corbett, Hubert Denise, Matthew Fraser, Alejandra Gonzalez-Beltran, Christopher Hunter, Philip Jones, Rasko Leinonen, Craig McAnulla, Eamonn Maguire, and Others. 2014. "EBI metagenomics—a new resource for the analysis and archiving of metagenomic data." *Nucleic Acids Research* 42 (D1). Oxford Univ Press: D600——D606.

Huson, Daniel H, Suparna Mitra, Hans-Joachim Ruscheweyh, Nico Weber, and Stephan C Schuster. 2011. "Integrative analysis of environmental sequences using MEGAN4." *Genome Research* 21 (9): 1552–60. doi:10.1101/gr.120618.111.

Huson, Daniel H., Alexander F. Auch, Ji Qi, and Stephan C. Schuster. 2007. "MEGAN analysis of metagenomic data." *Genome Research* 17 (3): 377–86. doi:10.1101/gr.5969107.

Huttenhain, R., M. Soste, N. Selevsek, H. Rost, a. Sethi, C. Carapito, T. Farrah, E. W. Deutsch, U. Kusebauch, R. L. Moritz, E. Nimeus-Malmstrom, O. Rinner, and R. Aebersold. 2012. "Reproducible Quantification of Cancer-Associated Proteins in Body Fluids Using Targeted Proteomics." *Science Translational Medicine* 4 (142): 142ra94–142ra94. doi:10.1126/scitranslmed.3003989.

Jablonski, D., and W. G. Chaloner. 1994. "Extinctions in the Fossil Record [and Discussion]." *Philosophical Transactions of the Royal Society B: Biological Sciences* 344 (1307): 11–17. doi:10.1098/rstb.1994.0045.

Jagtap, Pratik D., Alan Blakely, Kevin Murray, Shaun Stewart, Joel Kooren, James E. Johnson, Nelson L. Rhodus, Joel Rudney, and Timothy J. Griffin. 2015. "Metaproteomic analysis using the Galaxy framework." *Proteomics* 15 (20): 3553–65. doi:10.1002/pmic.201500074.

Juste, Catherine, David P Kreil, Christian Beauvallet, Alain Guillot, Sebastian Vaca, Christine Carapito, Stanislas Mondot, et al. 2014. "Bacterial protein signals are associated with Crohn's disease." *Gut* 63 (10): 1566–77. doi:10.1136/gutjnl-2012-303786.

Kanehisa, Minoru, and Susumu Goto. 2000. "KEGG: Kyoto Encyclopaedia of Genes and Genomes." *Nucl. Acids Res.* 28: 27–30. doi:10.1093/nar/28.1.27.

Kelchtermans, Pieter, Wout Bittremieux, Kurt De Grave, Sven Degroeve, Jan Ramon, Kris Laukens, Dirk Valkenborg, Harald Barsnes, and Lennart Martens. 2014. "Machine learning applications in proteomics research: How the past can boost the future." *Proteomics* 14 (4-5): 353–66. doi:10.1002/pmic.201300289.

Kent, W. J. 2002. "BLAT—The BLAST-Like Alignment Tool." *Genome Research* 12 (4): 656–64. doi:10.1101/gr.229202.

Kim, Dongjae, Aria S Hahn, Shang-ju Wu, Niels W Hanson, Kishori M Konwar, and Steven J Hallam. 2015. "FragGeneScan-Plus for scalable high-throughput short-read open reading frame prediction." In *IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 2015*, 1–8. doi:10.1109/cibcb.2015.7300341.

Knoll, a H, E J Javaux, D Hewitt, and P Cohen. 2006. "Eukaryotic organisms in Proterozoic oceans." *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences* 361 (1470): 1023–38. doi:10.1098/rstb.2006.1843.

Kolmeder, Carolin A., and Willem M. de Vos. 2014. "Metaproteomics of our microbiome — Developing insight in function and activity in man and model systems." *Journal of Proteomics* 97: 3–16. doi:10.1016/j.jprot.2013.05.018.

Kolmeder, Carolin A., Mark de Been, Janne Nikkilä, Ilja Ritamo, Jaana Mättö, Leena Valmu, Jarkko Salojärvi, Airi Palva, Anne Salonen, and Willem M. de Vos. 2012. "Comparative metaproteomics and diversity analysis of human intestinal microbiota testifies for its temporal stability and expression of core functions." *PLoS ONE* 7 (1). doi:10.1371/journal.pone.0029913.

Koonin, Eugene V, and Yuri I Wolf. 2010. "Constraints and plasticity in genome and molecular-phenome evolution." *Nature Reviews. Genetics* 11 (7): 487–98. doi:10.1038/nrg2810.

Linnaeus, C. 1758. *Systema naturae per regna tria naturae. 2 vols.* doi:10.1007/s13398-014-0173-7.2.

Liu, Bo, and Mihai Pop. 2011. "MetaPath: identifying differentially abundant metabolic pathways in metagenomic datasets." *BMC Proceedings* 5 (Suppl 2): S9. doi:10.1186/1753-6561-5-S2-S9.

Lowe, Christopher J. 2013. "The Cambrian Explosion The Construction of Animal Biodiversity." *SCIENCE* 340: 1170–1. doi:10.1126/science.1237431.

Lozupone, Catherine, and Rob Knight. 2005. "UniFrac: a new phylogenetic method for comparing microbial communities." *Applied and Environmental Microbiology* 71 (12): 8228–35. doi:10.1128/AEM.71.12.8228-8235.2005.

Luo, Chengwei, Luis M. Rodriguez-R, and Konstantinos T. Konstantinidis. 2014. "MyTaxa: An advanced taxonomic classifier for genomic and metagenomic sequences." *Nucleic Acids Research* 42 (8): 1–12. doi:10.1093/nar/gku169.

Ma, Bin, Kaizhong Zhang, Christopher Hendrie, Chengzhi Liang, Ming Li, Amanda Doherty-Kirby, and Gilles Lajoie. 2003. "PEAKS: powerful software for peptide de novo sequencing by tandem mass spectrometry." *Rapid Communications in Mass Spectrometry : RCM* 17 (20): 2337–42. doi:10.1002/rcm.1196.

MacLean, Brendan, Daniela M. Tomazela, Nicholas Shulman, Matthew Chambers, Gregory L. Finney, Barbara Frewen, Randall Kern, David L. Tabb, Daniel C. Liebler, and Michael J. MacCoss. 2010. "Skyline: An open source document editor for creating and analyzing targeted proteomics experiments." *Bioinformatics* 26 (7): 966–68. doi:10.1093/bioinformatics/btq054.

Mande, Sharmila S., Monzoorul Haque Mohammed, and Tarini Shankar Ghosh. 2012. "Classification of metagenomic sequences: Methods and challenges." *Briefings in Bioinformatics* 13 (6): 669–81. doi:10.1093/bib/bbs054.

Marshall, Charles R. 2006. "Explaining the Cambrian 'Explosion' of Animals." *Annual Review of Earth and Planetary Sciences* 34 (1): 355–84. doi:10.1146/annurev.earth.33.031504.103001.

Mayr, Ernst. 1982. *The Growth of Biological Thought: Diversity, Evolution, and Inheritance*. Vol. 1. doi:10.1016/0162-3095(84)90038-4.

McDonald, Daniel, Jose C Clemente, Justin Kuczynski, Jai Rideout, Jesse Stombaugh, Doug Wendel, Andreas Wilke, Susan Huse, John Hufnagle, Folker Meyer, Rob Knight, and J Caporaso. 2012. "The Biological Observation Matrix (BIOM) format or: how I learned to stop worrying and love the ome-ome." *GigaScience* 1 (1): 7. doi:10.1186/2047-217X-1-7.

Medini, Duccio, Claudio Donati, Hervé Tettelin, Vega Masignani, and Rino Rappuoli. 2005. "The microbial pan-genome." *Current Opinion in Genetics and Development* 15 (6): 589–94. doi:10.1016/j.gde.2005.09.006.

Mehlan, Henry, Frank Schmidt, Stefan Weiss, Julia Schüler, Stephan Fuchs, Katharina Riedel, and Jörg Bernhardt. 2013. "Data visualization in environmental proteomics." *Proteomics* 13 (18-19): 2805–21. doi:10.1002/pmic.201300167.

Mesuere, Bart, Griet Debyser, Maarten Aerts, Bart Devreese, Peter Vandamme, and Peter Dawyndt. 2015. "The Unipept metaproteomics analysis pipeline." *Proteomics* 15 (8): 1437–42. doi:10.1002/pmic.201400361.

Mesuere, Bart, Bart Devreese, Griet Debyser, Maarten Aerts, Peter Vandamme, and Peter Dawyndt. 2012. "Unipept: Tryptic peptide-based biodiversity analysis of metaproteome samples." *Journal of Proteome Research* 11 (12): 5773–80.

Mesuere, Bart, Felix Van der Jeugt, Bart Devreese, Peter Vandamme, and Peter Dawyndt. 2016a. “The unique peptidome: taxon-specific tryptic peptides as biomarkers for targeted metaproteomics.” *Proteomics*, no. (accepted with minor revisions).

Mesuere, Bart, Toon Willems, Felix Van der Jeugt, Bart Devreese, Peter Vandamme, and Peter Dawyndt. 2016b. “Unipept web services for metaproteomics analysis.” *Bioinformatics*, 1–2. doi:10.1093/bioinformatics/btw039.

Meyer, Folker, D Paarmann, M D’Souza, and Etal. 2008. “The metagenomics RAST server—a public resource for the automatic phylo- genetic and functional analysis of metagenomes.” *BMC Bioinformatics* 9: 386. doi:10.1186/1471-2105-9-386.

Miller, R.B. 1968. “Response time in man-computer conversational transactions.” In *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*, 267–77. doi:10.1145/1476589.1476628.

Mora, Camilo, Derek P. Tittensor, Sina Adl, Alastair G B Simpson, and Boris Worm. 2011. “How many species are there on earth and in the ocean?” *PLoS Biology* 9 (8). doi:10.1371/journal.pbio.1001127.

Moruz, Luminita, An Staes, Joseph M. Foster, Maria Hatzou, Evy Timmerman, Lennart Martens, and Lukas Käll. 2012. “Chromatographic retention time prediction for post-translationally modified peptides.” *Proteomics* 12 (8): 1151–9. doi:10.1002/pmic.201100386.

Nielsen, H Bjørn, Mathieu Almeida, Agnieszka Sierakowska Juncker, Simon Rasmussen, Junhua Li, Shinichi Sunagawa, Damian R Plichta, et al. 2014. “Identification and assembly of genomes and genetic elements in complex metagenomic samples without using reference genomes.” *Nature Biotechnology* 32 (8): 822–28. doi:10.1038/nbt.2939.

Olsen, Jesper V, Shao-En Ong, and Matthias Mann. 2004. “Trypsin cleaves exclusively C-terminal to arginine and lysine residues.” *Molecular & Cellular Proteomics : MCP* 3 (6): 608–14. doi:10.1074/mcp.T400003-MCP200.

Olson, Michael A, Keith Bostic, and Margo Seltzer. 1999. “Berkeley {DB}.” In *Proceedings of the Annual Conference on {USENIX} Annual Technical Conference*, 43–43. http://dl.acm.org/citation.cfm?id=1268708.1268751.

Oparin, Alexander. 1953. *Origin of Life*. Vol. 11. 4.

Pan, Sheng, Ru Chen, Randall E Brand, Sarah Hawley, Yasuko Tamura, Philip R Gafken, Brian P Milless, David R Goodlett, John Rush, and Teresa A Brentnall. 2012. “Multiplex Targeted Proteomic Assay for Biomarker Detection in Plasma: A Pancreatic Cancer Biomarker Case Study.” *JOURNAL OF PROTEOME RESEARCH* 11 (3): 1937–48. doi:10.1021/pr201117w.

Pasteur, Louis. 1862. “Mémoire sur les corpuscules organisés qui existent dans l’atmosphère: examen de la doctrine des générations spontanées.”

Pellicer, Jaume, Michael F. Fay, and Ilia J. Leitch. 2010. “The largest eukaryotic genome of them all?” *Botanical Journal of the Linnean Society* 164 (1): 10–15. doi:10.1111/j.1095-8339.2010.01072.x.

Pennisi, E. 2012. “ENCODE Project Writes Eulogy for Junk DNA.” *Science* 337 (6099): 1159–61. doi:10.1126/science.337.6099.1159.

Perkins, D N, D J Pappin, D M Creasy, and J S Cottrell. 1999. “Probability-based protein identification by searching sequence databases using mass spectrometry data.” *Electrophoresis* 20 (18): 3551–67. doi:10.1002/(SICI)1522-2683(19991201)20:18<3551::AID-ELPS3551>3.0.CO;2-2.

Pevtsov, Sergey, Irina Fedulova, Hamid Mirzaei, Charles Buck, and Xiang Zhang. 2006. “Performance evaluation of existing de novo sequencing algorithms.” *Journal of Proteome Research* 5 (11): 3018–28. doi:10.1021/pr060222h.

Prlić, Andreas, Andrew Yates, Spencer E. Bliven, Peter W. Rose, Julius Jacobsen, Peter V. Troshin, Mark Chapman, Jianjiong Gao, Chuan Hock Koh, Sylvain Foisy, Richard Holland, Gediminas Rimša, Michael L. Heuer, H. Brandstätter-Müller, Philip E. Bourne, and Scooter Willis. 2012. “BioJava: An open-source framework for bioinformatics in 2012.” *Bioinformatics* 28 (20): 2693–5. doi:10.1093/bioinformatics/bts494.

Qin, Junjie, Ruiqiang Li, Jeroen Raes, Manimozhiyan Arumugam, Kristoffer Solvsten Burgdorf, Chaysavanh Manichanh, Trine Nielsen, et al. 2010. “A human gut microbial gene catalogue established by metagenomic sequencing : Article : Nature.” *Nature* 464 (7285): 59–65. doi:10.1038/nature08821.

Renne, Paul R, Alan L Deino, F.J. Hilgen, Klaudia F Kuiper, Darren F Mark, William S Mitchell, Leah E Morgan, Roland Mundil, and Jan Smit. 2013. “Time scales of critical events around the Cretaceous-Paleogene boundary.” *Science (New York, N.Y.)* 339 (6120): 684–7. doi:10.1126/science.1230492.

Rho, Mina, Haixu Tang, and Yuzhen Ye. 2010. “FragGeneScan: Predicting genes in short and error-prone reads.” *Nucleic Acids Research* 38 (20): 1–12. doi:10.1093/nar/gkq747.

Riding, Robert. 1999. “The term stromatolite: towards an essential definition.” *Lethaia* 32 (January): 321–30. doi:10.1111/j.1502-3931.1999.tb00550.x.

Riehmann, Patrick, Manfred Hanfler, and Bernd Froehlich. 2005. “Interactive sankey diagrams.” In *Proceedings - IEEE Symposium on Information Visualization, INFO VIS*, 233–40. doi:10.1109/INFVIS.2005.1532152.

Rooijers, Koos, Carolin Kolmeder, Catherine Juste, Joël Doré, Mark de Been, Sjef Boeren, Pilar Galan, Christian Beauvallet, Willem M de Vos, and Peter J Schaap. 2011. "An iterative workflow for mining the human intestinal metaproteome." *BMC Genomics* 12: 6. doi:10.1186/1471-2164-12-6.

Rost, H., L. Malmstrom, and R. Aebersold. 2012. "A Computational Tool to Detect and Avoid Redundancy in Selected Reaction Monitoring." *Molecular & Cellular Proteomics* 11 (8): 540–49. doi:10.1074/mcp.M111.013045.

Schopf, J William, Anatoliy B Kudryavtsev, David G Agresti, Thomas J Wdowiak, and Andrew D Czaja. 2002. "Laser-Raman imagery of Earth's earliest fossils." *Nature* 416 (6876): 73–76. doi:10.1038/416073a.

Schwartz, M. 2001. "The life and works of Louis Pasteur." *Journal of Applied Microbiology* 91 (4): 597–601. doi:10.1046/j.1365-2672.2001.01495.x.

Segata, Nicola, Jacques Izard, Levi Waldron, Dirk Gevers, Larisa Miropolsky, Wendy S Garrett, and Curtis Huttenhower. 2011. "Metagenomic biomarker discovery and explanation." *Genome Biology* 12 (6): R60. doi:10.1186/gb-2011-12-6-r60.

Seifert, Jana, Florian Alexander Herbst, Per Halkjær Nielsen, Francisco J. Planes, Nico Jehmlich, Manuel Ferrer, and Martin Von Bergen. 2013. "Bioinformatic progress and applications in metaproteogenomics for bridging the gap between genomic sequences and metabolic functions in microbial communities." *Proteomics* 13 (18-19): 2786–2804. doi:10.1002/pmic.201200566.

Seshadri, Rekha, Saul a. Kravitz, Larry Smarr, Paul Gilna, and Marvin Frazier. 2007. "CAMERA: A community resource for metagenomics." *PLoS Biology* 5 (3): 0394–97. doi:10.1371/journal.pbio.0050075.

Song, Baoxing, Xiaoquan Su, Jian Xu, and Kang Ning. 2012. "MetaSee: An Interactive and Extendable Visualization Toolbox for Metagenomic Sample Analysis and Comparison." *PLoS ONE* 7 (11): 1–10. doi:10.1371/journal.pone.0048998.

Stonebraker, Michael, and Ariel Weisberg. 2013. "The VoltDB Main Memory DBMS." *IEEE Data Eng. Bull.*, 21–27. http://sites.computer.org/debull/a13june/voltdb1.pdf.

Tanaka, Koichi, Hiroaki Waki, Yutaka Ido, Satoshi Akita, Yoshikazu Yoshida, and Tamio Yohida. 1988. "Protein and polymer analyses up to m/z 100,000 by laser ionization time-of-flight mass spectrometry." *Rapid Communications in Mass Spectrometry* 2 (8): 151–53.

Tanca, Alessandro, Antonio Palomba, Massimo Deligios, Tiziana Cubeddu, Cristina Fraumene, Grazia Biosa, Daniela Pagnozzi, Maria Filippa Addis, and Sergio Uzzau. 2013. "Evaluating the impact of different sequence databases on metaproteome analysis: Insights from a lab-assembled microbial mixture." *PLoS ONE* 8 (12). doi:10.1371/journal.pone.0082981.

Tatusova, Tatiana, Stacy Ciufo, Scott Federhen, Boris Fedorov, Richard McVeigh, Kathleen O'Neill, Igor Tolstoy, and Leonid Zaslavsky. 2015. "Update on RefSeq microbial genomes resources." *Nucleic Acids Research* 43 (Database issue): D599–605. doi:10.1093/nar/gku1062.

The Gene Ontology Consortium. 2014. "Gene Ontology Consortium: going forward." *Nucleic Acids Research* 43 (D1): D1049–D1056. doi:10.1093/nar/gku1179.

The UniProt Consortium. 2014. "Activities at the Universal Protein Resource (UniProt)." *Nucleic Acids Research* 42 (Database issue): D191–8. doi:10.1093/nar/gkt1140.

The Uniprot Consortium. 2015. "UniProt: a hub for protein information." *Nucleic Acids Research* 43 (Database issue): D204–12. doi:10.1093/nar/gku989.

Tsilia, Varvara, Bart Devreese, Ilse De Baenst, Bart Mesuere, Andreja Rajkovic, Mieke Uyttendaele, Tom Van De Wiele, and Marc Heyndrickx. 2012. "Application of MALDI-TOF mass spectrometry for the detection of enterotoxins produced by pathogenic strains of the Bacillus cereus group." *Analytical and Bioanalytical Chemistry* 404 (6-7): 1691–1702.

Vandermarliere, Elien, Michael Mueller, and Lennart Martens. 2013. "Getting intimate with trypsin, the leading protease in proteomics." *Mass Spectrometry Reviews* 32 (6): 453–65. doi:10.1002/mas.21376.

Venter, J C, M D Adams, E W Myers, P W Li, R J Mural, G G Sutton, H O Smith, et al. 2001. "The sequence of the human genome." *Science (New York, N.Y.)* 291 (5507): 1304–51. doi:10.1126/science.1058040.

Verberkmoes, Nathan C, Alison L Russell, Manesh Shah, Adam Godzik, Magnus Rosenquist, Jonas Halfvarson, Mark G Lefsrud, Juha Apajalahti, Curt Tysk, Robert L Hettich, and Janet K Jansson. 2009. "Shotgun metaproteomics of the human distal gut microbiota." *The ISME Journal* 3 (2): 179–89. doi:10.1038/ismej.2008.108.

Vital, Marius, Adina Chuang Howe, and James M Tiedje. 2014. "Revealing the bacterial butyrate synthesis pathways by analyzing (meta)genomic data." *MBio* 5 (2): e00889. doi:10.1128/mBio.00889-14.

Vizcaíno, Juan Antonio, Richard G. Côté, Attila Csordas, José A. Dianes, Antonio Fabregat, Joseph M. Foster, Johannes Griss, Emanuele Alpi, Melih Birim, Javier Contell, Gavin O'Kelly, Andreas Schoenegger, David Ovelleiro, Yasset Pérez-Riverol, Florian Reisinger, Daniel Ríos, Rui Wang, and Henning Hermjakob. 2013. "The Proteomics Identifications (PRIDE) database and associated tools: Status in 2013." *Nucleic Acids Research* 41 (D1). doi:10.1093/nar/gks1262.

Wang, Rui, Yasset Perez-Riverol, Henning Hermjakob, and Juan Antonio Vizcaino. 2015. "Open source libraries and frameworks for biological data visualisation: A guide for developers." *Proteomics* 15 (8): 1356–74. doi:10.1002/pmic.201400377.

Watson, James D, and Francis H C Crick. 1953. "Molecular structure of nucleic acids." *Nature* 171 (4356): 737–38. doi:10.1097/BLO.0b013e3181468780.

Wheeler, David L, Deanna M Church, Ron Edgar, Scott Federhen, Wolfgang Helmberg, Thomas L Madden, Joan U Pontius, Gregory D Schuler, Lynn M Schriml, Edwin Sequeira, Tugba O Suzek, Tatiana A Tatusova, and Lukas Wagner. 2004. "Database resources of the National Center for Biotechnology Information: update." *Nucleic Acids Research* 32 (Database issue): D35–D40. doi:10.1093/nar/gkh073.

White, James Robert, Niranjan Nagarajan, and Mihai Pop. 2009. "Statistical methods for detecting differentially abundant features in clinical metagenomic samples." *PLoS Computational Biology* 5 (4): e1000352. doi:10.1371/journal.pcbi.1000352.

Wilmes, Paul, and Philip L. Bond. 2004. "The application of two-dimensional polyacrylamide gel electrophoresis and downstream analyses to a mixed community of prokaryotic microorganisms." *Environmental Microbiology* 6 (9): 911–20. doi:10.1111/j.1462-2920.2004.00687.x.

Wood, Derrick E, and Steven L Salzberg. 2014. "Kraken: ultrafast metagenomic sequence classification using exact alignments." *Genome Biology* 15 (3): R46. doi:10.1186/gb-2014-15-3-r46.

Wu, Cathy H, Rolf Apweiler, Amos Bairoch, Darren A Natale, Winona C Barker, Brigitte Boeckmann, Serenella Ferro, Elisabeth Gasteiger, Hongzhan Huang, Rodrigo Lopez, Michele Magrane, Maria J Martin, Raja Mazumder, Claire O'Donovan, Nicole Redaschi, and Baris Suzek. 2006. "The Universal Protein Resource (UniProt): an expanding universe of protein information." *Nucleic Acids Research* 34 (Database issue): D187–D191. doi:10.1093/nar/gkj161.

Yamada, Takuji, Ivica Letunic, Shujiro Okuda, Minoru Kanehisa, and Peer Bork. 2011. "IPath2.0: Interactive pathway explorer." *Nucleic Acids Research* 39 (SUPPL. 2): 1–4. doi:10.1093/nar/gkr313.

Yang, Youngik, and Shibu Yooseph. 2013. "SPA: A short peptide assembler for metagenomic data." *Nucleic Acids Research* 41 (8): 1–10. doi:10.1093/nar/gkt118.