2019

Methods in automated glycosaminoglycan tandem mass spectra analysis

BOSTON UNIVERSITY

GRADUATE SCHOOL OF ARTS AND SCIENCES

AND

COLLEGE OF ENGINEERING

Dissertation

METHODS IN AUTOMATED GLYCOSAMINOGLYCAN TANDEM MASS SPECTRA ANALYSIS

by

JOHN DANIEL HOGAN

B.A., Auburn University, 2007
M.S., University of Georgia, 2010

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

2019

Approved by

First Reader

Joseph Zaia, Ph.D.
Professor of Biochemistry

Second Reader

Cheng Lin, Ph.D.
Research Associate Professor of Biochemistry

This work is dedicated to all of the teachers I have had in my life, from Mrs. Ayers to Dr. Zaia. I wouldn't be here without your guidance and inspiration!

Acknowledgments

There are many people I need to acknowledge as instrumental to my Ph.D. journey. First of all, I give my sincerest thanks to the BU Bioinformatics Program for its guidance and devotion to its students. In particular, the administration – Mary Ellen Gipson-Fitzpatrick, Dave King, Caroline Lyman, and Johanna Vasquez – has been a source of unwavering support throughout my years in the program, and they have all helped with my sanity during these years.

I also would like to thank the Center for Biomedical Mass Spectrometry (CBMS) at the BU School of Medicine. In particular, I would like to give special thanks to Han Hu, Joshua Klein, Jiandong Wu, and William Hackett. Han was the caretaker for this project before I joined the CBMS, and his prior work has been extremely helpful for me as a foundation. Josh is an incredible programmer who also learned everything there is to know about glycomics; his broad expertise has been extraordinarily helpful and an inspiration for my time at BU. Jiandong has been a tremendous resource for GAG MS^2; not only has he generated the MS^2 data, he has also helped me understand the limitations and utility of the data. Finally, Will has been a newer member of CBMS and the BU Program in Bioinformatics, but he has still been a good sounding board for ideas and general commentary; I know that this project (and other glycomic bioinformatics projects) is in good hands moving forward. To the rest of the CBMS: I wouldn't be where I am today without you, and I wish you all the best of luck moving forward.

Of course, I would not be in this position without my thesis advisory committee. First of all, I would like to thank my original committee chair, Scott Mohr, for his guidance and motivation; Scott retired during my time at BU, but I learned much from him in our time together, and I cherish our conversations. Andrew Emili is

the newest member of my committee, but his expertise on both MS^2 and network analysis have been instrumental for my thesis. Tom Tullius took over for Dr. Mohr as both my committee chair and resident chemist, and has been a solid addition to my committee. Luis Carvalho has been the mathematics and statistics member of my committee, and his statistical and algorithmic suggestions for my projects were always spot on. My second reader, Cheng Lin, has helped me tremendously in terms of understanding the nuances of MS^2 and the various methods for fragmentation, and also in describing these different methods appropriately; I sincerely appreciate your guidance and ease of explanation, as they are crucial contributions to my research and work at BU. Finally, I need to acknowledge my advisor Joe Zaia. From the moment I joined the lab, you have made me feel at home, even when I felt like an outsider after changing labs mid-stream. Our meetings and discussions have been extraordinarily helpful both in terms of my understanding of GAG biochemistry and MS^2 and keeping me on the right path. I owe you my sincerest thanks, and wish you the very best in the future.

I sincerely need to thank my many Boston friends I made along my Ph.D. journey for helping me remain sane. From Jim Dundon and BO Tuesdays at Club Wilson to Justin Mann and snowball fights to Ania Tassinari and dance parties, to many, many more, you have all been tremendous friends while graduate school swirled around me. I also want to thank Wilson Ramiro, Joseph Rogers, Allyson Byrd, Heather Selby, Jessica Keenan, Lingqi Luo, Adam Labadorf, Arjan van der Velde, the brothers Wixom – Andy and Nick – Allen Miller, Charlie Lissandrello, Dan Lancour, Evan Appleton, Devanshi Patel, and many others for always being supportive and there for me. From the bottom of my heart, thank you for being there for me.

Since my youth, my family has been there for me, and I would be remiss to forget mentioning them. To my brother Adam, thank you for always being there for a laugh

and for rooting for an inferior NFL team (#riseup). To my mother, thank you for your constant affirmations and your sincere belief that I can achieve anything. To my father, thank you for your quick wit and your less quick history lessons. And to my extended family and my new in-laws, thank you for the many years of love you have bestowed on me, and your faith in my endeavors.

Finally, I need to give sincere gratitude and love to my wife, Brenna. Since we started at BU together, we have always had an undeniable chemistry and have brought out the best in each other. My admiration and love for you has only grown, and I cannot express enough how much you have meant to me over these years. With you in my corner, I know I can do anything I set my mind to. I love you more than you will ever know, and I can't wait to see what the future has in store for us.

METHODS IN AUTOMATED GLYCOSAMINOGLYCAN TANDEM MASS SPECTRA ANALYSIS

JOHN DANIEL HOGAN

Boston University, Graduate School of Arts and Sciences and College of Engineering, 2019

Major Professor: Joseph Zaia, Ph.D.
Professor of Biochemistry

ABSTRACT

Glycosylation is the process by which a glycan is enzymatically attached to a protein, and is one of the most common post-translational modifications in nature. One class of glycans is the glycosaminoglycans (GAGs), which are long, linear polysaccharides that are variably sulfated and make up the glycan portion of proteoglycans (PGs). PGs are located on the cellular surface and in the extracellular matrix (ECM), making them important molecules for cell signaling and ligand binding. The GAG sulfation sequence is a determining factor for the signaling capacity of binding complexes, so accurate determination of the sequence is critical. Historically, GAG sequencing using tandem mass spectrometry (MS^2) has been a difficult, manual process; however, with the advent of faster computational techniques and higher-resolution MS^2, high-throughput GAG sequencing is within reach.

Two steps in the pipeline of biomolecule sequencing using MS^2 are discovery and interpretation of spectral peaks. The discovery step traditionally is performed using methods that rely on the concept of averagine, or the average molecular building block for the analyte in question. These methods were developed for protein sequencing, but

perform considerably worse on GAG sequences, due to the non-uniform distribution of sulfur atoms along the chain and the relatively high isotope abundance of ^{34}S. The interpretation step traditionally is performed manually, which takes time and introduces potential user error. To combat these problems, I developed GAGfinder, the first GAG-specific MS^2 peak finding and annotation software. GAGfinder is described in detail in Chapter 2.

Another step in MS^2 sequencing is the determination of the sequence using the found MS^2 fragments. For a given GAG composition, there are many possible sequences, and peak finding algorithms such as GAGfinder return a list of the peaks in the MS^2 mass spectrum. The many-to-many relationship between sequences and fragments can be represented using a bipartite network, and node-ranking techniques can be employed to generate likelihood scores for possible sequences. I developed a bipartite network-based sequencing tool, GAGrank, based on a bipartite network extension of Google's PageRank algorithm for ranking websites. GAGrank is described in detail in Chapter 3.

Contents

List of Tables

List of Figures

List of Abbreviations

Asn	.....	Asparagine
AUC	.....	Area Under the Curve
CID	.....	Collision Induced Dissociation
CS	.....	Chondroitin Sulfate
CSV	.....	Comma-separated Variable
DAG	.....	Directed Acyclic Graph
dp	.....	Degree of Polymerization
DS	.....	Dermatan Sulfate
ECM	.....	Extracellular Matrix
EDD	.....	Electron Detachment Dissociation
EID	.....	Experimental Isotopic Distribution
ES	.....	Enrichment Score
ETD	.....	Electron Transfer Dissociation
ExD	.....	Electron Activated Dissociation
FDR	.....	False Discovery Rate
FN	.....	False Negative
FP	.....	False Positive
FPI	.....	Free Proton Index
GAG	.....	Glycosaminoglycan
Gal	.....	Galactose
GalN	.....	Galactosamine
GalNAc	.....	*N*-acetylgalactosamine
GlcA	.....	Glucuronic Acid
GlcN	.....	Glucosamine
GlcNAc	.....	*N*-acetylglucosamine
GO	.....	Gene Ontology
GSEA	.....	Gene Set Enrichment Analysis
HCD	.....	Higher-energy Collisional Dissociation
Hep	.....	Heparin
HS	.....	Heparan Sulfate

ID	.....	Isotopic Distribution
IdoA	.....	Iduronic Acid
KNN	.....	k-Nearest Neighbors
KS	.....	Keratan Sulfate
KSI	.....	Keratan Sulfate Type I
KSII	.....	Keratan Sulfate Type II
KSIII	.....	Keratan Sulfate Type III
LC-MS^2	.....	Liquid Chromatography-Tandem Mass Spectrometry
LFQ	.....	Label-free Quantiication
m/z	.....	Mass-to-Charge Ratio
MALDI	.....	Matrix Assisted Laser Desorption/Ionization
Man	.....	Mannose
MS^2	.....	Tandem Mass Spectrometry
NETD	.....	Negative Electron Transfer Dissociation
NRE	.....	Non-Reducing End
PC	.....	Principal Component
PCA	.....	Principal Components Analysis
PCR	.....	Polymerase Chain Reaction
PerfScore	.....	Performance Score
PG	.....	Proteoglycan
PNP	.....	4-nitrophenol
P-R	.....	Precision-Recall
Pro	.....	Proline
PTM	.....	Post-Translational Modification
RE	.....	Reducing End
SA	.....	Simulated Annealing
Ser	.....	Serine
SLRP	.....	Small Leucine-Rich Proteoglycan
S/N	.....	Signal-to-Noise Ratio
TDA	.....	Target-Decoy Approach
Thr	.....	Threonine
TID	.....	Theoretical Isotopic Distribution
TP	.....	True Positive

Chapter 1

Review of Bioinformatics Techniques for Glycan Sequencing Using Tandem Mass Spectrometry

1.1 Biological Background

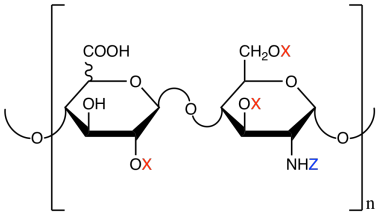
The traditional central dogma of molecular biology states that DNA is transcribed into RNA, which is then translated into protein. This offers a tidy, but incomplete, picture of what happens at the molecular level in living systems. Research over the past century or so has unearthed more intricate molecular machinery that affects this process, including epigenetic modifications and post-translational modifications (PTMs), among others. Epigenetic modifications include DNA methylation, where a methyl group is added to cytosine bases in a DNA sequence, and histone modifications, where the histone protein is chemically altered via methylation, acetylation, or other means. Both of these types of epigenetic modifications alter gene expression by affecting the binding capability of transcriptional proteins. PTMs are chemical modifications that change how proteins interact with other molecules. Varieties of PTMs include phosphorylation, ubiquitination, and glycosylation, among many oth-

ers. Glycosylation is the process of covalently binding a carbohydrate polysaccharide – termed a glycan – to a protein sequence, and is the general focus of this dissertation.

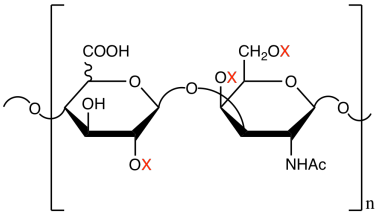
Protein glycosylation is classified according to the type of linkage to the amino acid side chains: *O*-linked glycans, *N*-linked glycans, and glycosaminoglycans (GAGs). *O*-linked glycans bind to the oxygen atom in a serine (Ser) or threonine (Thr) amino acid residue, and the presence of a proline (Pro) residue at either the -1 or +3 position of the protein sequence promotes *O*-linked glycosylation. There are eight core structures of mucin-type *O*-glycans, all of which are initiated with an *N*-acetylgalactosamine (GalNAc) residue. However, for other *O*-glycans, there are no consensus sequences, and there is considerable diversity in the number and type of structures in this class of glycan. *N*-linked glycans bind to the nitrogen atom in the side chain of the asparagine (Asn) residue of a protein's sequon, which is Asn-X-Ser or Asn-X-Thr, where X is any amino acid except Pro. All animal *N*-linked glycans contain a branched pentasaccharide core that is comprised of two β1→4-linked *N*-acetylglucosamine (GlcNAc) residues – known as chitobiose – and three mannose (Man) residues. *N*-linked glycan biosynthesis proceeds after formation of this core, and results in one of three classes of branched glycans, depending on when the process is halted. High Man *N*-linked glycans consist exclusively of Man residues attached to the chitobiose core, complex *N*-linked glycans proceed further through biosynthesis and replace the non-core Man residues with other monosaccharide residues, and hybrid *N*-linked glycans have had biosynthesis arrested during the process of converting from high Man to complex.

GAGs are the third class of glycans, and they are long, linear polysaccharides that typically comprise the glycan portion of proteoglycans (PGs), although they also can be found attached to the cellular membrane or free-floating in the extracellular matrix (ECM). GAGs covalently bind to proteins at Ser residues via a linker polysaccharide, which depends on the GAG class. There are three main classes of sulfated GAGs that

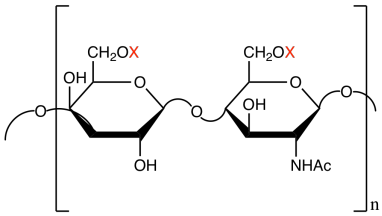
Heparan Sulfate/Heparin



Chondroitin Sulfate/Dermatan Sulfate



Keratan Sulfate



X = H or SO_3
Z = H or Ac or SO_3

Figure 1·1: Characteristic disaccharides including linkage information and sulfation location information for each sulfated GAG class.

each have a characteristic repeating disaccharide unit consisting of a uronic acid or galactose residue and a hexosamine residue, as shown in Figure 1·1. The disaccharide units are distinguished by their compositions, linkages, and sulfation locations. For each class of GAG, individual sequences vary by their degree of polymerization (dp), amount and location of sulfation, and uronic acid epimerization.

1.1.1 Heparan Sulfate and Heparin

The heparan sulfate (HS) disaccharide unit has more possible sulfation positions than any other GAG class, and thus more variability. The uronic acid, which is either iduronic acid (IdoA) or glucuronic acid (GlcA), can be sulfated at the 2-*O* position. The hexosamine, which is always glucosamine (GlcN), can be sulfated at the 2-*N*, 3-*O*, and 6-*O* positions, and can be acetylated at the 2-*N* position. Rarely, the 2-*N* position of the GlcN is neither sulfated nor acetylated, resulting in a free amine. HS chains organize into domains of heavy sulfation, heavy non-sulfation, and hybrid domains. The HS disaccharide unit linkage is [-4GlcA/IdoAβ1-4GlcNα1-]. HS is closely related to heparin (Hep), which follows the disaccharide structure shown in Figure 1·1, though with slight differences. Hep has a higher IdoA content, has more 2-*N* sulfation of the GlcN residue, and is overall more sulfated than HS. Furthermore, Hep is found in granulated hematopoeitic lineage cells. Mast cells are used for industrial production of biopharmaceutical heparin. HS, by contrast, is expressed on plasma membrane-bound proteins on most cell surfaces and in ECM PGs.

PGs that carry covalently bound HS GAGs include transmembrane syndecans and glypicans and ECM PGs such as perlecan and agrin. Syndecans are involved in growth factor binding [1–5], ECM adhesion [6–8], cell-cell adhesion [7, 9–11], and tumor suppression [12]. Glypicans are involved in developmental morphogenesis [13, 14] and regulation of cell signaling [15, 16]. Perlecan is a vascular ECM PG that is involved in endothelial barrier function [17, 18]. Agrin has several various roles in embryonic development [19, 20].

1.1.2 Chondroitin Sulfate and Dermatan Sulfate

The disaccharide unit for chondroitin sulfate (CS) and the closely related dermatan sulfate (DS) is slightly different from that of HS. Once again, the uronic acid can be either IdoA or GlcA – CS contains only GlcA, while DS can contain either – and can be sulfated at the 2-*O* position. The hexosamine, which is always galactosamine (GalN), can be sulfated at the 4-*O* or 6-*O* positions. Unlike HS, the GalN in CS and DS is always *N*-acetylated. The CS/DS disaccharide linkage is [-4GlcA/IdoAβ1-3GalNAcβ1-]. Historically, DS was referred to as chondroitin sulfate B, but that terminology has fallen out of favor.

CS and DS are the bound GAGs for a wide range of PGs. Decorin is a small cellular PG that is associated with fibrillogenesis [21–24]. Biglycan is a small ECM PG that is associated with bone mineralization [25]. Versican is a hyalectan PG that is found in ECM molecular lattices in many connective tissues and thereby regulates avaibility of growth factors to the cell surface [26–28]. Neurocan is a hyalectan found only in neural tissue, dysregulation of the expression of which is associated with bipolar disorder [29]. Aggrecan is the largest hyalectan that provides the swelling pressure for viscoelastic connective tissues including cartilage, tendon, and intervertebral disk [30, 31]. Brevican is a central nervous system hyalectan that is found in perineuronal nets, the dysregulaton of which is associated with neurodegenerative and neuropsychiatric disorders [32–34]. CS, like HS, is a GAG bound to perlecan, which is described above.

1.1.3 Keratan Sulfate

The keratan sulfate (KS) disaccharide unit is essentially a sulfated lactosamine unit found on *N*- or *O*-glycans. Because of this, some glycoproteins are considered part-time PGs if they become sulfated on lactosamine residues in certain biological context. Rather than a uronic acid residue, it has a galactose (Gal) residue, which can be

sulfated at the 6-O position. The hexosamine residue is GlcN, and can be sulfated at the 6-O position. Like CS/DS, the GlcN is always N-acetylated. The KS disaccharide linkage is [-3Galβ1-4GlcNAcβ1-]. There are three classes of KS: keratan sulfate type I (KSI), keratan sulfate type II (KSII), and keratan sulfate type III (KSIII). These classes differ in how they link to protein structures. KSI is linked to a high mannose N-glycan precursor oligosaccharide linker that is N-linked to an Asn residue. KSII is linked to the GalNAc residue of a mucin core 2 linker that is itself O-linked to a Ser or Thr residue. KSIII is linked to a mannose residue that is O-linked to a Ser residue.

KS PGs include the small leucine-rich PG (SLRP) family members fibromodulin, associated with collagen fiber assembly [35], and lumican, which is most closely associated with corneal transparency and also has a role in epithelial cell migration and tissue repair [36]. Dysregulation of another SLRP family member, keratocan, plays a role in the rare congenital corneal disease cornea plana 2 [37, 38].

1.2 Tandem Mass Spectrometry

Tandem mass spectrometry (MS^2) is an analytical chemistry technique by which biomolecular structure can be determined. There are multiple stages to the process, illustrated in Figure 1·2. For biomolecules, the analyte is ionized using a gentle ionization technique such as electrospray ionization (ESI) or matrix-assisted laser desorption/ionization (MALDI). For GAGs, ESI is most often used because it introduces ions into the gas phase with minimal vibrational excitation. The resulting ion(s), termed the precursor ion(s), are resolved according to mass-to-charge ratio (m/z) by a mass analyzer. In this thesis, I consider mass spectral data generated using a Fourier transform ion cyclotron resonance (FTICR) mass analyzer whereby an ion image current in the time domain is transformed into the frequency domain

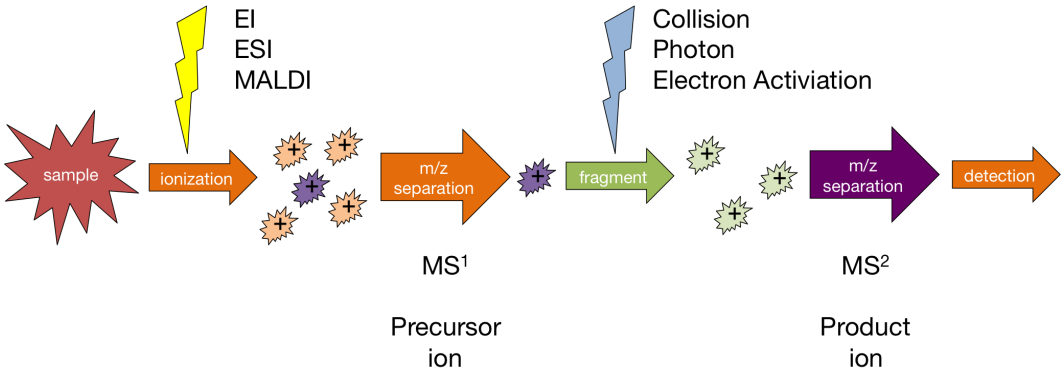


Figure 1·2: Illustration of the stages of tandem mass spectrometry, from injection to detection. Adapted from `https://en.wikipedia.org/Tandem_mass_spectrometry`.

and then converted into m/z. An ion corresponding to a GAG saccharide of interest is then selected for dissociation. This thesis focuses on data generated using electron activated dissociation (ExD). The m/z values, ion charge, and isotope pattern define the compositions of molecular fragments created during the MS^2 dissociation experiment.

Most modern vendor software for MS^2 analysis comes with a built-in peak finding algorithm, and there are numerous algorithms that have been developed for structure determination for specific purposes. The typical steps for a MS^2 experiment include spectrum analysis and peak interpretation. Spectrum analysis involves both detecting and annotating fragment peaks. Peak interpretation includes reverse engineering the biomolecular structure based on the fragment peaks in the spectrum. Historically, these steps would be done manually in a tedious and time-consuming manner. In the proteomics domain, however, with the sequencing of the genomes of many organisms, it is possible to use database searching to identify the peptides from which tandem mass spectra were produced.

MS^2 is a valuable tool in glycomics research. Often, changes in glycan structure either cause or result from different biological processes, including disease progression.

For instance, due to the role of many glycans in cellular signaling, glycan structure is a key player in various cancers [39–44]. The ability to reconstruct glycan sequences is a key benefit to using MS^2 for glycan analysis. MS^2 can also be used to estimate glycan abundances, which allows for differential expression analysis. The ability to test for changes to glycan structure and relative abundance among different datasets allows for robust study of these mechanisms, and exhibits the utility of MS^2 as a key method for glycan research.

1.3 Computational Techniques for Branched Glycan Sequencing

Traditional sequencing methods, such as those used for nucleic acid or protein sequencing, are inappropriate for branched glycans (i.e., *O*-glycans and *N*-glycans), due to their branched nature and extra complexity arising from different linkage possibilities. Furthermore, glycan biosynthesis is not template-driven, so there is not a glycan code that is analogous to the genetic code of codons being translated into amino acids. Therefore, more nuanced methods are required for automated sequencing of glycans. There are numerous algorithms for branched glycan sequencing, which fall into two categories based on the approach used: database-assisted methods and *de novo* methods. This subsection will contain a review of some of the techniques used in these methods for branched glycan sequencing. However, it is not meant to be a complete enumeration of all existing methods for branched glycan sequencing; rather, it is meant to highlight some of the approaches used for this problem.

1.3.1 Database-Assisted Sequencing for Branched Glycans

One category of branched glycan sequencing methods is database-assisted sequencing. Curated databases such as GlyTouCan [45, 46], UniCarbKB [47], GlycomeDB

[48], BCSDB [49], and GlycoEpitope [50], among others, store glycan structures and their associated biochemical information found in the literature. Most methods that rely on a structure database utilize one of these, depending on the exact question they seek to answer. For instance, BCSDB is solely concerned with bacterial carbohydrate structures, and GlycoEpitope stores information about the relationship between antibodies and glycan antigens.

GlycoMod [51] was published in 2001 with the goal of determining glycan compositions for either *N*-linked or *O*-linked glycans, given user-input constraints. GlycoMod relies on a list of existing glycan compositions and masses to match masses observed in a mass spectrum. GlycoMod also links to UniProt databases [52] for matching peptide sequences. A downside for GlycoMod is that it only returns glycan compositions without any linkage information. This is due to GlycoMod being developed for MS^1 data rather than MS^2 data, which would provide fragmentations of the glycan.

GlycosidIQ [53] was a glycan sequencing software suite published in 2004 that matched observed MS^2 fragment masses to a set of masses corresponding to resulting fragments of an *in silico* fragmentation of known glycan structures from a database. GlycosidIQ made use of a commercial glycan structure database, GlycoSuiteDB [54], that contained curated and annotated glycan structures. GlycosidIQ used two scoring algorithms to rank possible structures, structure-independent segmentation and structure-relative correspondence. GlycosidIQ is no longer maintained.

Also in 2004, Lohmann and von der Lieth published GlycoFragment and GlycoSearchMS [55], two tools developed for glycan sequencing. The two tools worked in tandem: GlycoFragment generated potential fragments for a mass spectrum, while GlycoSearchMS compared them to those found in a database of theoretically generated spectra, SweetDB [56]. Candidate structures were scored by how well the theoretical and observed peaks overlapped given a particular error tolerance. Neither

GlycoFragment nor GlycoSearchMS is still maintained.

GlyDB [57] was published in 2007 for glycopeptide glycan annotation and is based on the SEQUEST peptide sequence database search. GlyDB was developed for low-energy CID MS^2 that produces abundant B- and Y-ion fragments. GlyDB also implements a linear string representation for branched glycans, although it does not distinguish between isomeric monosaccharides. GlyDB is no longer maintained.

GlycoPep Grader (GPG) [58] was a glycoproteomics software suite published in 2012 that used a target-decoy approach (TDA) for estimating false discovery rates (FDRs) for potential glycopeptides given MS^2 data. GPG evaluated both the peptide and glycan portion of glycopeptides in its implementation for sequencing. The structure database used is uploaded by the user or selected at runtime. GPG is no longer maintained.

GlycoPep Detector (GPD) [59] was a software suite developed by the same group as GPG that was published in 2013 and was designed exclusively for electron transfer dissociation (ETD) MS^2 spectra. They used a set of experimentally validated *N*-linked glycan spectra to train their algorithm on expected fragmentation patterns, and used this information to guide its sequencing scoring. Also similar to GPG, GPD used a TDA for estimating FDRs for candidate structures. GPD is no longer maintained.

GlycoMaster DB [60] was published in 2014 and is software for identifying intact *N*-linked glycopeptides given high-throughput MS^2 spectra. GlycoMaster DB searches both peptide and glycan databases separately to find the best match for the glycopeptide of interest. GlycoMaster DB expects higher-energy collisional dissociation (HCD) spectra for glycan fragmentation and ETD spectra for peptide sequences. GlycoMaster DB uses GlycomeDB for searching glycan structures.

Mayampurath *et al.* published a computational method, GlycoFragwork [61], for

identifying intact glycoproteins in 2014 that has separate scoring algorithms depending on the method of dissociation used. GlycoFragwork uses TDA for FDR estimation among potential structures. GlycoFragwork performs better when multiple modes of fragmentation are combined in an experiment, resulting in complementary data about the structure.

Another method for identifying intact N-linked glycopeptides is pGlyco [62]. pGlyco was published in 2016 and uses complementary HCD-MS^2 and collision induced dissociation (CID)-MS^2 data to sequence the glycan portion. A novel TDA using a finite mixture model is used for FDR estimation for potential glycan sequences. pGlyco also uses MS^3 for peptide backbone sequencing.

1.3.2 *De Novo* Sequencing for Branched Glycans

Sequencing methods that do not rely on a pre-existing database or library generated at run time are classified as *de novo* sequencing methods. These techniques rely on knowledge about the types of structure being analyzed without comparing to known, annotated structures. There are numerous different approaches to sequencing without databases.

The web application saccharide topology analysis tool (STAT) [63] was published in 2000 and considers all possible structures that match a given mass. Users select the composition that they feel is most likely, and STAT generates possible structures for that composition that match prior biosynthetic rules. STAT is limited in its scope to a size of ten monosaccharides, due to the numerical explosion associated with generating all possible structures.

StrOligo [64, 65] was published in 2002 and uses dynamic programming to build glycan structures based on MS^2 spectra. StrOligo builds a relationship tree that represents linkages between monosaccharides and proposes structures that match the tree. The proposed structures follow biosynthetic rules for the glycan class in ques-

tion. For instance, for *N*-glycans, StrOligo knows that the base of the tree will be a branched trimannosyl core. StrOligo is no longer maintained.

GLYCH [66] was published in 2005 and makes use of the cross-ring and double-fragmentation cleavages generated in high-energy collisional dissociation methods. Many database-assisted sequencing methods only considered glycosidic bond fragments for determining the sequence, but this leaves out a wealth of information about the structure. For branched glycans, cross-ring cleavage fragments can localize linkage information. GLYCH uses neighboring fragment information to build its most likely structure. GLYCH is no longer maintained.

In 2009, Peltoniemi *et al.* [67] published a branch-and-bound method for *de novo* glycan sequencing for intact glycopeptides. Like some of the above database-assisted intact glycopeptide methods, this method utilizes MS^2 at different collision energies to generate complementary information about the peptide sequence and the glycan sequence. The branch-and-bound algorithm is used to explore the search space of possible *N*-glycans without full enumeration, and candidate sequences are scored by how likely they would be to generate the spectra in question. This method is no longer maintained.

Böcker *et al.* published an algorithm in 2011 [68] that is an exact algorithm for quickly generating glycan tree structures that match spectra. They focus away from *N*-glycans, but with the caveat that it works on *N*-glycans if biological knowledge restricts the search space for potential structures. This method considers glycan topologies as rooted tree structures, restricting possibilities using biosynthetic information. The method also restricts the number of peaks to consider to a user-input value. There is no extant web application that uses the algorithm.

Dong et al. published a method in 2015 for building a branched glycan ontology using a novel directed acyclic graph (DAG) method for representing the glycan [69].

While this distinction may seem minor, encoding structures this way allows for the sequencing problem to be broken up into subgraph units, and the sequence is built iteratively from the "bottom up." This technique speeds up analysis over existing methods, which can be slow, depending on the amount of solution space the algorithm searches.

In 2016, Sun *et al.* published a heuristic algorithm for *de novo* glycan sequencing [70] that gives greater weight to MS^2 fragment peaks found in the higher end of the m/z dimension. This allows for a sort of top down sequencing algorithm, though it is not to be confused with top-down proteomics. Like most other *de novo* sequencing methods, they built the possible sequences as tree structures.

Recently, Hong *et al.* published GlycoDeNovo [71], which uses an interpretation graph to build a glycan topology representation. GlycoDeNovo's key innovation is its IonClassifier method, which distinguishes B- and C-ions from others via machine learning on training data. GlycoDeNovo ranks candidate topologies by their cumulative IonClassifier scores. In contrast to other algorithms, which run in exponential time, GlycoDeNovo runs in polynomial time, providing a considerable runtime reduction.

1.4 Computational Techniques for Glycosaminoglycan Sequencing

GAG structures are different from branched glycans, mostly due to their linear structure, repeating backbone disaccharide, and the importance of modifications. Therefore, the methods used for sequencing GAGs will be different than those used for branched glycans. This subsection will be a brief review of some of the methods developed for GAG sequencing using MS^2. Due to the relative paucity of GAG sequencing methods to date, compared to methods for branched glycan sequencing, this

subsection will not be broken up into separate parts.

The first effort to sequence GAGs was the heparin/HS oligosaccharide sequencing tool (HOST), published in 2005 [72]. HOST was a software application that made use of HS disaccharide composition information and MS^2 fragmentation information for enzymatically digested heparin using Microsoft Excel. This allowed for broad sequencing to be performed (i.e., the number of modifications on each monosaccharide), but not site-specific modification information.

In 2008, Tissot *et al.* published an extension [73] of GlycoWorkbench [74] for semiautomatic interpretation of GAG MS^2 data. This software includes a visual editor for glycan structures, GlycanBuilder, that allows users to build candidate structures. The software then performs *in silico* fragmentation of the candidate structures in order to generate all possible fragments for each of them. Now, the user can compare the spectra to the possible fragments for the candidate structures.

Spencer *et al.* published a method in 2010 for GAG structure prediction using disaccharide composition information and selective lyase digestion [75]. The stated goal for this method was HS domain prediction, and they employed a modular, three-step algorithm toward this end. The first step, chainmaker, builds candidate HS chains based on user inputs such as the length of the chain, disaccharide composition information, and the order of lyase digestion. The second step, chainbreaker, digests the potential chains *in silico* to see if the resulting disaccharide compositions match with the experimental ones. The final step, chainsorter, organizes the candidate structures into user-defined domains. While this method was an important step toward automated HS sequencing, it did not utilize MS^2 data, and only considered disaccharide information.

In 2014, Hu *et al.* published the first *de novo* sequencing algorithm for HS MS^2 data, HS-SEQ [76]. This approach takes as input a MS^2 peak list and information

about the HS backbone and returns positional modification probabilities for each HS modification location. HS-SEQ uses a spectrum graph model to map the path from one sequence terminus to the other using the found peaks. HS-SEQ assigns a confidence value and a uniqueness value to fragments based on how they fit into the existing structure, and builds the consensus sequence using higher confidence fragments first. The main drawback of HS-SEQ is that it does not produce a complete sequence, but rather a backbone with modification probabilities.

GAG-ID [77] was published in 2015 for high-throughput HS identification using liquid chromatography-tandem mass spectrometry (LC-MS^2). GAG-ID uses a multivariate hypergeometric distribution to place peaks in low-intensity, medium-intensity, and high-intensity bins. GAG-ID then compares the MS^2 spectra to *in silico* fragmentation of an exhaustive list of candidate structures to score their likelihood. A drawback of GAG-ID is that it requires an extensive chemical workup, including permethylation of N-acetyl groups and replacing sulfate groups with deuterated acetate. This reduces the problem of sulfate loss but introduces many more wet lab steps. The authors of GAG-ID published a multivariate mixture model for estimating the accuracy of GAG-ID's identifications [78]. They used an expectation-maximization algorithm to separate correct identifications from incorrect identifications, and concluded that GAG-ID is accurately identifying HS structures in their data.

As of this writing, the most recent published method for GAG sequencing is a genetic algorithm for moving through the space of possible structures [79]. Like most genetic algorithms, this method consists of three steps: initialization, crossover, and mutation. In the initialization step, two random GAG structures are generated for the given composition. In the crossover step, the two sequences exchange a modification that they do not have in common. In the mutation step, a modification is moved along the chain to a non-modified position for each sequence. These steps are repeated until

a stopping criterion is met, which is typically when there is little or no change in the fitness parameter, which is based on how the fragments of the potential structure compare to the experimental data.

1.5 Future Directions in Glycosaminoglycan Sequencing

Great strides have been made in the realm of GAG structure determination over the past twenty years, but a consensus has not been reached regarding the best approach. Disagreements on how to handle sulfate loss, whether to adduct cations, and the exact nature of the methods are some of the culprits. Furthermore, the field is narrow in scope, meaning fewer groups focus on it than other fields such as proteomics, genomics, or even *N*-linked and *O*-linked glycomics. That said, the current options for GAG sequencing all have positive qualities, and all are useful for predicting GAG sequences given MS^2 data.

GAG sequencing method performance is directly related to its inputs, and the most crucial of these is the MS^2 data. Improvements in MS^2 methods will only improve GAG sequencing methods, and can do so in a number of ways. First of all, while ExD fragmentation produces a large complement of fragments, it still does not sufficiently define all positions of backbone modification. This is the cause for some of the ambiguity found in the existing approaches to GAG sequence determination. Were a complete fragmentation of the sequence possible, any of these methods would be able to select the correct sequence every time. Second, MS^2 resolution and noise reduction are constantly improving. Most current methods are developed for high-resolution data, and operate with a very small error window for peak matching. Distinguishing the signal from the noise is imperative in many statistical methods, and GAG sequencing is no different. Having a higher signal-to-noise ratio will result

in more found fragments, which will in turn improve performance for sequencing methods. Third, reducing or removing false peak identifications will help reduce ambiguity in identifications. In any mass spectra, there are a number of peaks that do not correspond to the structure being analyzed. This could be due to sample contamination, co-isolation, or noise spikes. These peaks harm the methods' ability to identify the correct structure given the found fragments.

An overlooked aspect of GAG sequencing is the determination of C5 epimers on uronic acid residues. One unfortunate aspect of MS^2 as a sequencing technique is that it does not distinguish these epimers, since it is based on mass. Several groups have shown that there are fragment ions in MS^2 spectra that are diagnostic for GlcA or IdoA, including cross-ring cleavages near the uronic acid C5 [80–82]. However, this information is hardly automated, and deals with specific ions for specific sequences, so generalizing the results for other uronic acid sites is unlikely to work. There are other techniques, however, that could make progress in this sub-area of GAG sequencing. For instance, a decision tree could be employed based on curated GAG sequence MS^2 data that predicts whether each uronic acid in a GAG sequence is GlcA or IdoA. This would most likely find more diagnostic ions for general GAG conditions.

Finally, while numerous machine learning and statistical methods have been employed for this problem, there are still others that have not been tried. The realms of bioinformatics, probabilistic modeling, and machine learning are vast and constantly growing, meaning more and more techniques are available for GAG sequencing. Only after a more exhaustive utilization of these techniques can consensus of the best method be approached.

Chapter 2

Software for Peak Finding and Elemental Composition Assignment for Glycosaminoglycan Tandem Mass Spectra

2.1 Introduction

Glycosaminoglycans (GAGs) exist either as the glycan portion of proteoglycans (PGs) or as extracellular matrix (ECM) polysaccharides. The three classes of sulfated GAGs, heparan sulfate (HS), chondroitin sulfate (CS), and keratan sulfate (KS), are characterized by their long, linear chain, a repeating disaccharide unit (specific to each GAG class), and variable patterns of sulfation and acetylation. Due to their locations on the cell surface and in the ECM, as well as their sequence variation, they interact with many growth factors and growth factor receptors and therefore modulate cellular signaling and signal transduction pathways [83, 84]. Furthermore, spatial and temporal regulation of the structures of GAGs characterizes physiology and pathophysiology in eukaryotes. For instance, cancer cells remodel HS chains in their microenvironments to avoid immune system targeting and allow proliferation [85]. In the motor neuron-

degenerative disease amyotrophic lateral sclerosis, KS sulfation has been shown to correlate with disease progression [86]. Indeed, GAG expression is required for embryonic development [87], and GAGs are required for the proper functioning of all mammalian biological systems [83]. Clearly, assigning GAG sequences from tandem mass spectral data is necessary to establish their roles in diverse disease mechanisms.

Tandem mass spectrometry (MS^2) entails isolating a precursor ion in the first stage, and dissociating it in subsequent stages. Manual interpretation of tandem mass spectra is tedious, time-consuming, and subjective. The first step of interpretation is to assign the m/z and charge states for product ions. Once this is done, neutral masses and isotope compositions can be assigned. Once these assignments are made, an algorithm can be used to identify the GAG sequence [76].

Wolff and colleagues first applied electron activated dissociation methods to GAG oligosaccharides, using both electron detachment dissociation (EDD) [80] and negative electron transfer dissociation (NETD) [88]. More recently, Huang and colleagues showed the effectiveness of electron activated dissociation for minimizing sulfate loss during HS mass spectrometry experiments [89]. Resulting tandem mass spectra after electron activated dissociation are extremely rich in that they contain a large number of product ions with varying charge states and isotope patterns. In the proteomics domain, several computational methods for automatic recognition of isotopic patterns and assignment of charge states and neutral mass values have been developed, including THRASH [90], Decon2LS [91], and MS-Deconv [92], among others. These methods assume product ion isotopic distributions will match the pattern produced by the molecule's average building block, or averagine; however, performance for GAG saccharide tandem mass spectra is inadequate, due to the variable levels of sulfation along their chains and the relatively abundant ^{34}S isotope. Figure 2·1 shows two examples of the large difference in the expected isotopic distributions of non-sulfated

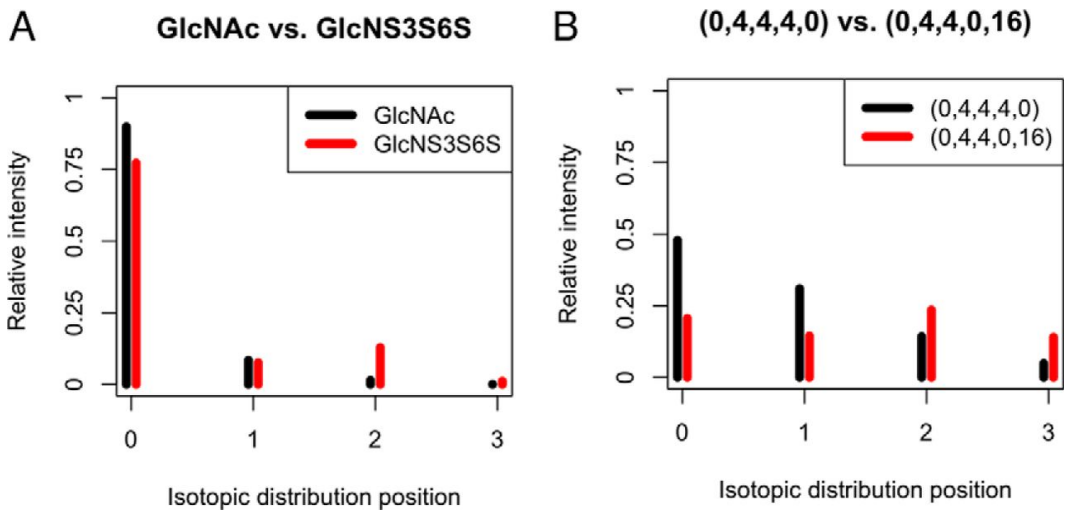


Figure 2·1: Comparison of expected isotopic distributions for oligosaccharides with varying sulfation. A. Expected isotopic distribution of non-sulfated *N*-acetylglucosamine (GlcNAc) compared with 3,6-*O*-sulfated, *N*-sulfated glucosamine (GlcNS3S6S). B. Expected isotopic distribution of a non-sulfated octasaccharide with acetyl groups at all four *N* positions compared with a hexadecasulfated octasaccharide with sulfate groups at every possible position. Notice the higher intensity at the A+2 peak for each fully sulfated oligosaccharide; for the octasaccharide, the A+2 peak has the highest intensity, making monoisotopic peak detection more difficult. Intensity is relative to the total intensity for the whole isotopic distribution. Key for octasaccharide: [ΔHexA, HexA, GlcN, Ac, SO_3].

and fully sulfated GAG fragments. Plainly, there is no GAG averagine that would accurately recover the correct monoisotopic peak for each fragment, and that leads to incorrect and missing assignments. Averagine-based approaches also do not assign elemental compositions for monoisotopic ions, a step necessary for interpretation of GAG saccharide tandem mass spectra. We sought to solve these problems.

Previous work in GAG tandem mass spectra analysis and annotation has typically been a step in a further sequencing project. For instance, Yu and colleagues recently sequenced the dermatan sulfate (DS) chain of the pericellular PG decorin using a genetic algorithm based on known sulfate modification information from disaccharide analysis, but mentioned in-house data interpretation software in passing [93]. And two

GAG sequencing efforts from Chiu and colleagues, GAG-ID [77] and a multivariate mixture model to estimate identification accuracy [78] represent recent attempts at automated GAG sequencing using a weighted hypergeometric distribution to match spectra to potential sequences. However, these papers both describe a method that only considers high intensity peaks, rather than full isotopic distributions, and their method requires an intense experimental workup for chemical derivatization that replaces sulfate groups with heavy isotope acetyl groups.

Averagine-based deisotoping and charge state deconvolution algorithms were developed to circumvent the combinatorial explosion of the number of possible protein sequences as the length of the chain increases. Due to this expansion, brute force methods searching all possible proteins and protein product ions are not feasible. While the number of possible GAGs also increases exponentially as a function of chain length, the rate of increase is much lower. Figure 2·2 shows the $\log_{10}$ of the number of possible structures of unmodified proteins, HS GAG saccharides, CS GAG saccharides, and KS GAG saccharides, as a function of the length of the chain. Notice how the slopes for each GAG class are much smaller than the slope for proteins, and consider how many more protein structures are possible when post-translational modifications are included. Given the reduced search space and the variable sulfation along GAG chains, we developed a brute force product ion search algorithm using the Python programming language, GAGfinder, for MS^2 of GAG saccharides of a given composition. GAGfinder iterates through every possible fragment of a GAG composition at multiple charge states and tests its theoretical isotopic distribution against the observed spectral pattern. GAGfinder is available for download at http://www.bumc.bu.edu/msr/software. This paper describes the steps in GAGfinder and its performance as a means to identify the GAG monoisotopic product ions, charge states, and neutral mass values versus an averagine-based peak finding

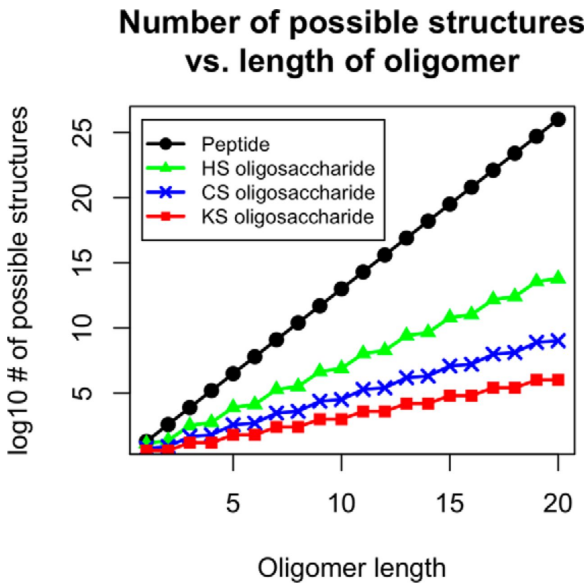


Figure 2·2: Workflow for GAGfinder. The steps in GAGfinder's algorithm.

algorithm.

2.2 Experimental Procedures

2.2.1 GAGfinder Overview

A flowchart of the steps GAGfinder can be viewed in Figure 2·3. The details of each step are described below. The term "product ion" will be used to refer to ions observed in tandem mass spectra. The term "fragment" will be used to refer to theoretical GAG saccharide substructures in a database.

Inputs – There are a number of required and optional inputs for GAGfinder to return accurate results. The spectrum data must be in the mzML file format [94]; the raw data can be converted using any format conversion tool, such as MSConvert [95] or compassXport (Bruker Daltonics, Inc.). Other required inputs include the GAG class, the precursor m/z, the precursor charge, and the output format for the results. Either the top percentile or the top N results can be returned, but not

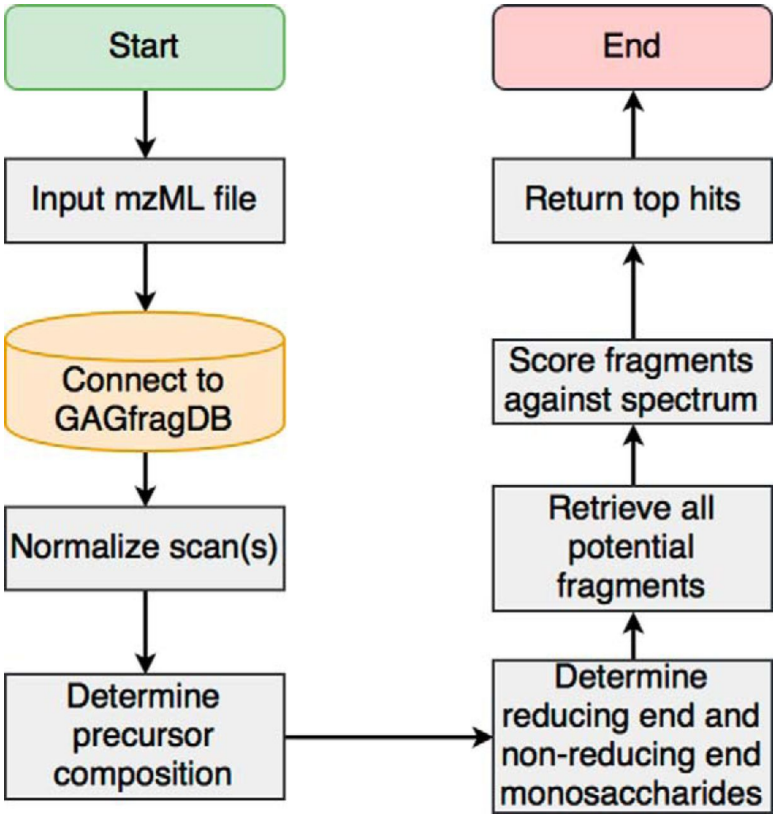


Figure 2·3: Plot of $\log_{10}$ of the number of possible structures given oligomer length. The number of unmodified protein sequences of a given oligomer length grows at a much faster rate than those of HS, CS, or KS. The slower combinatorial growth rate allows GAGfinder's brute force search to be feasible.

both. Optional inputs include the reducing-end derivatization formula (if any), the adducted metal and the number of adducts (if there is metal adduction), the NETD cation reagent (if NETD), a user-specified internal precision for mapping fragments to isotopic distributions, a Boolean value for whether noise has already been removed from the spectrum, and the number of labile sulfate losses to consider. These inputs are arguments for the GAGfinder command line program.

Step 1: Load mzML file and connect to GAG fragment database – The first step of GAGfinder is connecting to GAGfragDB, the database developed in SQLite for easy storing and retrieval of all possible fragments of a precursor composition up to hexadecamer. There are 4,150 unique compositions, 65,664 fragments, and 17,156,928 precursor-fragment mappings in GAGfragDB. The composition with the most possible fragments – (1, 7, 8, 4, 15) with a key of (ΔHexA, HexA, HexN, Ac, SO_3) – has 21,299 child fragments associated with it in HS. GAGfragDB includes a controlled vocabulary designed to give each fragment a unique text identifier that does not assume anything about the structure of the precursor or the fragment. In other words, a fragment that has one composition but could be a terminal fragment or any number of internal fragments will have only one identifier. Figure B·1 shows the relational schema for GAGfragDB. The connection to GAGfragDB is established by the Python sqlite3 module. After connecting to GAGfragDB, GAGfinder loads the mzML file into Python using the pymzML module [96]. The pymzML module has a number of spectrum processing methods, including centroiding peaks, finding peaks in the spectrum within a particular error tolerance, and a number of others.

Step 2: Normalize scan(s) and remove noise – Once the tandem mass spectral data have been loaded into Python, GAGfinder normalizes and averages the scans of the data file using the total ion current (TIC). GAGfinder first divides each scan in the file by the summed TIC intensity and then calculates the average over all scans.

This step prevents any of the scans from biasing the results over the rest of the scans, and is performed using methods in the pymzML package. After normalizing the scans, GAGfinder removes noise from the spectrum, if the spectrum has not already been denoised by the user prior to runtime. GAGfinder uses an implementation of the noise reduction algorithm MasSPIKE [97].

Step 3: Determine precursor composition – Given the precursor m/z and charge, the neutral mass of the precursor can be calculated, and based on this and the GAG class, the precursor composition can be determined. GAGfinder considers metal adduction and reducing end derivatization information in order to calculate the neutral mass matching the composition in GAGfragDB. GAGfinder selects the composition with the neutral mass closest to the calculated precursor mass as the precursor composition.

Step 4: Determine reducing end and non-reducing end monosaccharides – In order to reduce the search space as much as possible, GAGfinder attempts to determine the monosaccharides at each precursor saccharide terminus. There are several cases in which this is possible, and Figure 2·4 shows the decision tree for determining this. First, if the non-reducing end is an unsaturated uronic acid (in the cases of CS and HS saccharides generated by polysaccharide lyase enzyme digestion), GAGfinder first assumes that the reducing end monosaccharide is a hexuronic acid if the precursor contains an odd number of monosaccharides, and a hexosamine if the precursor contains an even number of monosaccharides. If this is not the case, then GAGfinder checks whether there is an unequal number of the parts of the repeating disaccharide for the current GAG class. If the number is unequal, then whichever monosaccharide there is more of will be on both the non-reducing and reducing end. If the number is equal, then GAGfinder cannot assign the end fragments and must search through the entire search space.

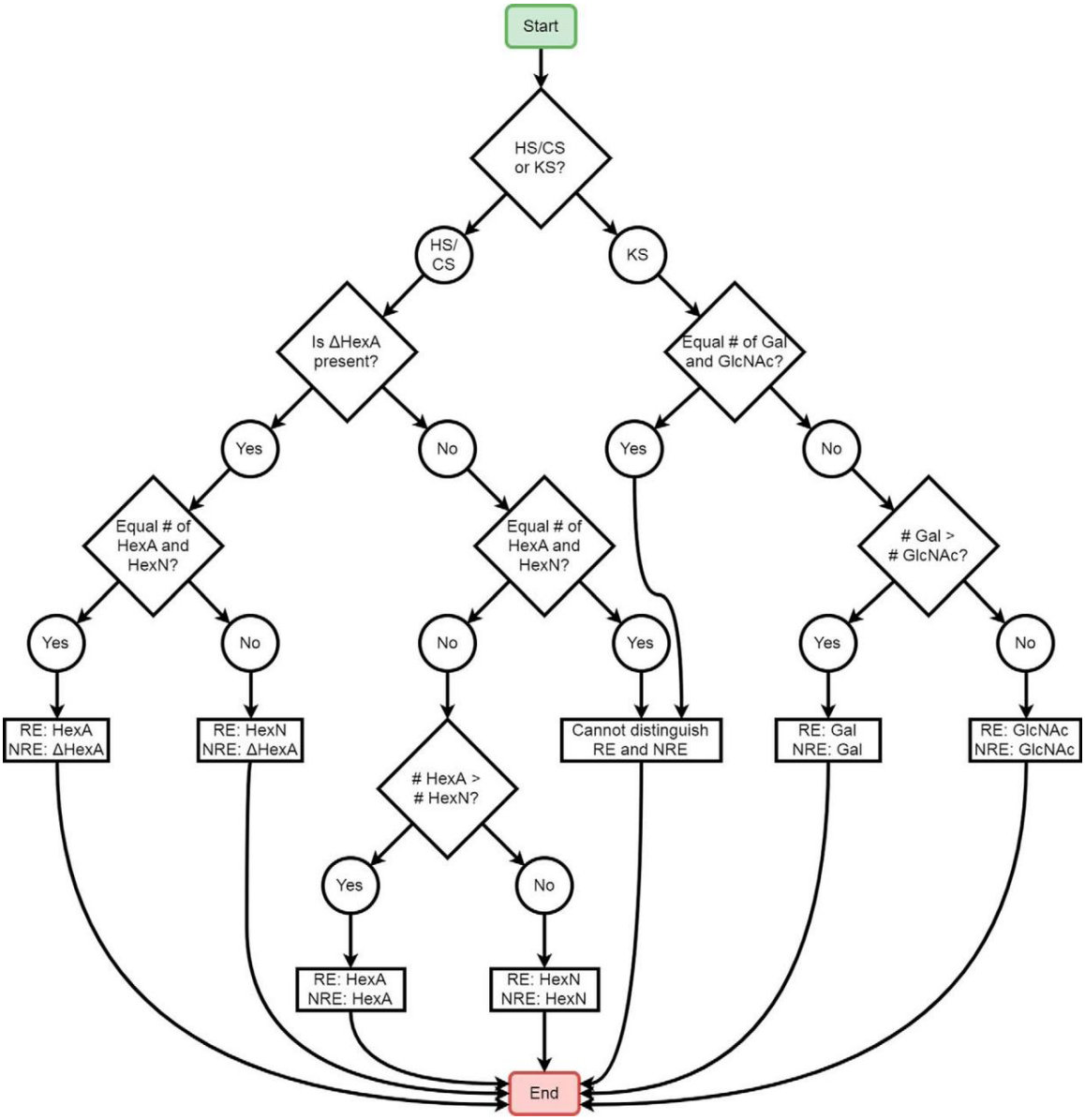


Figure 2·4: Flowchart describing steps in determining terminal sugars. In several cases, GAGfinder can determine the reducing and non-reducing end sugars based on biosynthetic rules. In cases where the sugars cannot be distinguished from the composition, both monosaccharides of the class of GAG are considered as the terminal sugars. RE = reducing end; NRE = non-reducing end.

Step 5: Retrieve and modify all theoretical fragments for the precursor – Next, GAGfinder retrieves every possible fragment for the current precursor from GAGfragDB. The possible fragments stored in GAGfragDB include glycosidic bond cleavages and all cross-ring cleavages except for those involving cleavage of adjacent bonds. Figure B·2 shows each cross-ring cleavage GAGfinder considers. GAGfragDB stores the theoretical fragments as neutral masses without considering sulfate losses or any other modification information, so GAGfinder must modify and search each fragment in order to maximize spectrum coverage. For each fragment, the modifications included are water loss (for glycosidic fragments only), hydrogen loss (up to 2), sulfate loss (up to the amount designated by the user), and reducing end derivatization (if any). This information is used to determine whether a given fragment corresponds to the reducing terminus. Product ions that have the same chemical composition are merged. For every combination of these modifications, the fragments are pushed through the algorithm.

Step 6: Score each theoretical fragment – Once all of the theoretical fragments have been retrieved and modified as need be, they are scored against the tandem mass spectrum. GAGfinder considers charge states from -1 to that of the precursor ion plus one for each fragment. The decision to use the charge state of the precursor ion plus one for the upper bound rather than that of the precursor ion is due to two main reasons. First, the number of product ions with the same charge state as the precursor is a small percentage of all of the product ions, meaning including this charge state in GAGfinder's searching would find only a few more product ions while introducing more false positives. Second, many of the product ions with the same charge state as the precursor are actually derivatives of the precursor, meaning they provide no additional structural information. A theoretical relative isotopic distribution (TID) is calculated for each fragment using the BRAIN algorithm [98],

which employs polynomial expansion and applies the Newton-Girard theorem and Viète's formulae to this end. Once the TID is calculated, GAGfinder searches the tandem mass spectrum for product ion peaks at the m/z values of the TID within either a user-specified error tolerance or the default error tolerance of 20 parts-per-million (ppm), storing them as the experimental isotopic distribution (EID). The EID is then divided by the sum of its intensities so that it is also a relative distribution. GAGfinder employs a G-test of goodness-of-fit to determine how similar the EID is to the TID. Equation 2.1 shows the expression for the G score, where i is the index of each peak in the matched isotopic distributions. According to the G-test, the G score follows a chi-squared distribution under the null hypothesis that the EID has the same distribution as the TID, and so can be used to compute p-values. This way, a lower G score yields a higher p-value and thus represents a better fit.

$$G = 2 \sum_i EID_i \ln \left(\frac{EID_i}{TID_i} \right) \tag{2.1}$$

Step 7: Rank product ions by G score and return top hits – Once all theoretical fragments have been scored for goodness-of-fit, they are ranked by increasing G score. Depending on whether the user requested the top percentile or top N results, those results are saved into an output file. The output file contains the fragment m/z, charge, intensity, annotation(s), G score, and error in ppm.

2.2.2 Data Acquisition and Preprocessing

We chose ten synthetic GAG standards to demonstrate the effectiveness of GAGfinder (Figure 2·5). These standards were chosen due to their range of modification distribution and precursor charges. Compounds 1 and 10 were synthesized as described [99]. Compound 2 was a generous gift from Prof. Jian Liu, University of North Carolina, Chapel Hill. Compound 3 was purchased from New England Biolabs (An-

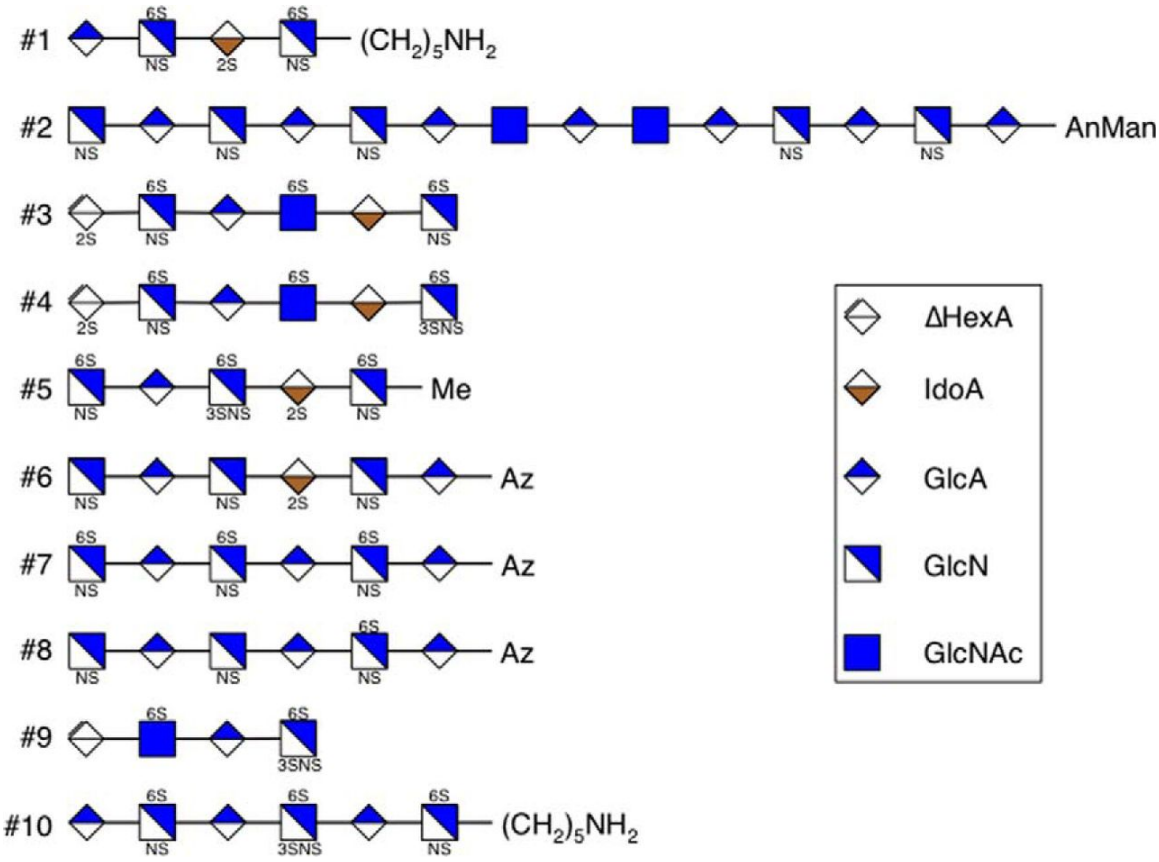


Figure 2·5: Structures of the ten synthetic standards used for testing purposes. #1 has charge state of 4- and dissociation method of NETD. #2 has charge state of 8- and dissociation method of NETD. #3 has charge state of 5- and dissociation method of NETD. #4 has charge state of 4- and dissociation method of EDD. #5 has charge state of 6- and dissociation method of EDD. #6 has charge state of 4- and dissociation method of NETD. #7 has charge state of 4- and dissociation method of NETD. #8 has charge state of 3- and dissociation method of EDD. #9 has charge state of 3- and dissociation method of NETD. #10 has charge state of 4- and dissociation method of EDD. These standards were selected randomly because of their range of modifications, length, and different dissociation methods.

dover, MA). Compound 5 was purchased as Arixtra pharmaceutical preparation and desalted by size exclusion chromatography. Compounds 4, 6, 7, and 8 were acquired through a publicly available set of HS standard saccharides funded by the NIH and maintained by the Zaia laboratory (http://www.bumc.bu.edu/zaia/gag-synthetic-saccharides-available/). Compound 9 was isolated from porcine intestinal mucosa as described [100]. These were subjected to electron detachment dissociation (EDD) or negative electron transfer dissociation (NETD) using a Bruker solariX 12T FTMS instrument. For each saccharide, GAGfinder was run retrieving 100% of tested fragments, allowed for two sulfate losses, and used the default error of 20 ppm when mapping fragments to isotopic distributions. For saccharides 1-5, noise was not previously removed, so GAGfinder implemented MasSPIKE to remove noise. For saccharides 6-10, noise was previously removed. While in principle GAGfinder can handle all classes of GAGs, we show results for HS saccharides for the present work. Details regarding the tandem mass spectrometric acquisition methods can be found in Hu, *et al.* [76]. Raw data files were converted to mzML format for input into GAGfinder by either MSConvertGUI version 3.0.5084 [92] or compassXport command line utility 3.0.13 (Bruker Daltonics, Inc.). The mass spectrometry glycomics data have been deposited to the ProteomeXchange Consortium via the PRIDE [101] partner repository with the dataset identifier PXD009101.

We first sought to demonstrate the ability of GAGfinder to identify product ion isotope clusters and charge states. To do this, we generated a list of product ions using a traditional averagine-based method (the SNAP peak finder in Bruker DataAnalysis 4.2) versus that for GAGfinder. In order to retrieve every product ion SNAP identified, we set the quality factor threshold at 0, the signal-to-noise ratio (S/N) threshold at 1, the relative intensity threshold (base peak) at 0%, and the absolute intensity threshold at 0. For each GAG saccharide tested, we set the maximum charge

state to the absolute value of the precursor charge state minus one, so that SNAP would behave comparatively to GAGfinder. We set the repetitive building block to $C_6H_{11.375}N_{1.125}O_{9.5}S_{1.5}$, as used in previous methods [102]. SNAP returned a matrix with columns for m/z, charge, intensity, resolving power, and quality factor.

2.2.3 Method Comparison

In order to judge GAGfinder's performance in assigning tandem mass spectral monoisotopic product ions and charge states, we employed two separate statistical methods. Each method required unbiased expert manual selection of monoisotopic product ion peaks to serve as the set of true positives. In both methods we had GAGfinder return scores for 100% of the tested theoretical fragments in order to ensure maximum spectral coverage. The first method compared the GAGfinder performance against that of a random selection of monoisotopic product ions. The second compared GAGfinder's performance to that of an averagine-based peak finding algorithm.

The first method for judging GAGfinder's performance was a permutation test that gauged GAGfinder's performance in selecting true positive product ion peaks compared against random selection of product ion peaks. First, we calculated a performance score (PerfScore) for the GAGfinder results using the equation

$$PerfScore = \sum_j G_j Hit_j \tag{2.2}$$

where j is the index of the current product ion, G_j is the G score for fragment j, and

$$Hit_j = \begin{cases} 1, & \text{if product ion } j \text{ is a "real" hit} \\ 0, & \text{if product ion } j \text{ is not a "real" hit} \end{cases} \tag{2.3}$$

Once we calculated the performance score for the GAGfinder results, we permuted the *Hit* vector 10,000 times and recalculated the performance score for each permutation.

Since G scores are smaller for better fits, a smaller performance score represents a better performance. The performance scores of the 10,000 permutations represent a background distribution for performance against which we compared the GAGfinder performance score. We plotted GAGfinder's performance score against the background distribution and recorded its rank among all of the permuted performance scores.

The second method for testing GAGfinder's performance was a binary classifier evaluation that compared the GAGfinder performance versus that of an averagine based algorithm, SNAP. Precision-recall (P-R) curves show how the classifier's precision and recall change as the classifier's threshold is changed, and the area under the curve (AUC) represents the classifier's performance. Precision is defined as

$$Precison = \frac{TP}{TP + FP} \tag{2.4}$$

where TP stands for true positives and FP stands for false positives, and recall, also known as sensitivity, is defined as

$$Recall = \frac{TP}{TP + FN} \tag{2.5}$$

where TP stands for true positives and FN stands for false negatives. A perfect classifier has a precision and a recall of 1, and therefore, the closer the P-R AUC is to 1, the better the classifier has performed. The complete results for GAGfinder were generated by requesting 100% of the product ions tested.

For GAGfinder results, we generated the vector of precision and recall values by ordering the results by G score in ascending order and calculating the precision and recall of GAGfinder at each G score threshold. Similarly, for SNAP, we ordered the results by quality factor in descending order and calculated its precision and recall at each quality factor threshold. Because the number of true positive peaks is limited

to those fragment masses extracted from GAGfragDB, GAGfinder identifies fewer monoisotopic peaks and charge states than does an averagine-based algorithm. In order to compare the effectiveness of the peaks assigned in common by both algorithms, we removed peaks that were not searched by GAGfinder. These peaks were likely due to fragmentation or chemistry that GAGfinder does not consider.

2.3 Results

2.3.1 GAGfinder Performance Compared to Random Sampling

For each of the ten GAG saccharide tandem mass spectra tested, the GAGfinder performance score significantly outperformed that of the permutations. Table 2.1 compares GAGfinder's performance score versus the mean and standard deviation of the 10,000 random permutations for each saccharide; the distribution plots for each saccharide can be seen in Figure B·3. For every compound, the PerfScore for GAGfinder was in the top ten lowest scores, and for seven of the compounds, the PerfScore for GAGfinder was lower than every permutation's PerfScore. This indicated that GAGfinder produced a better performance than a random selection. Furthermore, the GAGfinder PerfScore was at least three standard deviations lower than the average of the permutations for every saccharide, signifying significant outperformance compared to a random selection of peaks. There was no correlation between the PerfScores for GAGfinder and the means and standard deviations of the permutations, the dissociation method, or the precursor charge state. This indicated a lack of bias for the GAGfinder algorithm. We concluded based on these numbers that GAGfinder significantly outperforms a random selection of peaks.

Table 2.1: Performance scores for GAGfinder compared to the mean and standard deviation of the 10,000 permutations for each of the ten synthetic compounds. In each case, GAGfinder's PerfScore was lower than at least 99.9% of the permutations', indicating a better performance.

Compound	Precursor z	Dissociation Method	GAGfinder PerfScore	Permutation PerfScore Mean	Permutation PerfScore Std. Dev.	Rank
#1	4-	NETD	32.502	47.072	2.983	2/10,000
#2	8-	NETD	60.866	112.712	6.176	1/10,000
#3	5-	NETD	63.966	83.238	5.487	3/10,000
#4	4-	EDD	58.422	89.935	6.000	1/10,000
#5	6-	EDD	60.991	74.966	4.583	9/10,000
#6	4-	NETD	58.492	103.262	7.304	1/10,000
#7	4-	NETD	37.769	89.721	6.411	1/10,000
#8	3-	EDD	31.853	65.747	4.741	1/10,000
#9	3-	NETD	14.390	40.784	4.904	1/10,000
#10	4-	EDD	66.811	118.989	7.333	1/10,000

2.3.2 GAGfinder Performance Compared to Averagine-Based Peak Finding

Table 2.2 compares the P-R curve AUCs for each spectrum for GAGfinder versus SNAP, including summary statistics. The P-R curves for each spectrum are shown in Figure B·4. The average GAGfinder P-R AUC is higher than the average SNAP P-R AUC, while the median GAGfinder P-R AUC is almost equal to the median SNAP P-R AUC. For seven of the ten spectra, GAGfinder has a higher P-R AUC. Of these seven, four were generated by NETD, while the other three were generated by EDD, and there is no correlation between charge state and performance difference between GAGfinder and SNAP, indicating a lack of bias in the performance of each. These numbers show that GAGfinder identifies monoisotopic peaks and charge states with similar accuracy as does the averagine-based SNAP algorithm. We note again that GAGfinder assigns the elemental compositions for all identified monoisotopic peaks.

Table 2.2: Area under the curve (AUC) of precision-recall (PR) curves for GAGfinder analysis results compared to those from SNAP.

Compound	Precursor z	Dissociation Method	GAGfinder AUC	SNAP AUC
#1	4-	NETD	0.681	0.765
#2	8-	NETD	0.315	0.100
#3	5-	NETD	0.612	0.229
#4	4-	EDD	0.688	0.652
#5	6-	EDD	0.611	0.536
#6	4-	NETD	0.628	0.574
#7	4-	NETD	0.635	0.735
#8	3-	EDD	0.716	0.682
#9	3-	NETD	0.792	0.619
#10	4-	EDD	0.646	0.703
		Mean	0.632	0.560
		Std. Dev.	0.125	0.222
		Mean	0.641	0.636

2.3.3 Runtime Numbers for GAGfinder

GAGfinder tracks and reports the length of runtime for each analysis. The amount of time required for GAGfinder to search each fragment varies based on a variety of factors, but the two that affect runtime the most are the number of possible fragments and whether or not the noise was removed prior to analysis. Table 2.3 shows GAGfinder's runtime for each saccharide, as well as the total number of fragments for that composition and charge state combination and whether or not the data was pre-processed. As can be seen, analyzing a spectrum without noise removed greatly increases the runtime. This is not due to GAGfinder's noise removal step taking an inordinate amount of time, but rather due to the larger number of data points to average across scans. For instance, samples 1-5 did not have noise removed prior to analysis, leaving that step for GAGfinder, which slowed down runtime. However, samples 6-10 did have noise removed prior to analysis, and their faster runtime shows it.

Table 2.3: Runtime for GAGfinder analysis for each saccharide.

Compound	Precursor z	Dissociation Method	Runtime (s)	# of tested fragments/# of possible fragments	Noise removed?
#1	4-	NETD	124.450	122/4,089	No
#2	8-	NETD	176.900	827/99,120	No
#3	5-	NETD	139.566	177/12,256	No
#4	4-	EDD	76.679	252/9,147	No
#5	6-	EDD	71.833	172/6,870	No
#6	4-	NETD	1.740	413/7,398	Yes
#7	4-	NETD	1.885	391/8,373	Yes
#8	3-	EDD	1.724	257/4,932	Yes
#9	3-	NETD	1.653	160/2,524	Yes
#10	4-	EDD	1.830	428/8,379	Yes

2.4 Discussion

Here we have presented GAGfinder, the first GAG-specific isotopic distribution finding software for high resolution tandem mass spectra. GAGfinder uses a targeted, brute force approach to search observed product ions against a set of theoretical fragments calculated based on the precursor ion exact mass, composition based on GAG biosynthesis rules, and expected NETD and EDD tandem mass spectrometry dissociation patterns. The software is easy to use on any operating system and outperforms traditional peak finding software that was designed for peptide fragments. For this manuscript, GAGfinder was run as a command line utility on a MacBook Pro, and all tandem mass spectrometric data are available on the PRIDE Proteomics IDEntifications archive. While the software is currently only available in command line form, a web application and interface is currently under development and will be available soon.

We tested GAGfinder on the EDD and NETD spectra of a diverse set of synthetic GAGs and showed that it accurately and consistently returns valid fragments for the precursor being tested. GAGfinder consistently scored true positive fragments better than false fragments across all tested GAGs, and performed comparably to traditional peak finding methods. Unlike traditional peak finding methods, GAGfinder assigns elemental compositions to the monoisotopic product ions that are essential for assigning the saccharide structure. While we tested GAGfinder exclusively on high resolution spectra in the negative ion mode, the software was designed in principle to handle any resolution level in either the negative or positive ion mode. For low resolution spectra, we hypothesize that the G scores for assigned monoisotopic product ions will be worse than with high resolution data; however, this is due to the whole distribution of G scores shifting, and we anticipate that the correct IDs will still be found at or near the top of the ranked list of G scores.

While GAGfinder succeeds at identifying product ions that fall within the set defined in the GAGfragDB, it does not identify product ions that arise from undefined dissociation processes. Such undefined processes include rare dissociation patterns, a charge state equal to or higher in absolute value than that of the precursor, and random instrument noise. In these cases, traditional methods will have a greater likelihood of identifying m/z values and charge states but will not identify the elemental composition. Furthermore, these ions are not actually useful for GAG structure determination, which is the ultimate goal of GAG sequencing. While it is possible to add rare dissociation processes to the GAGfragDB, this would increase search space size at the expense of algorithm run time.

An interesting case where GAGfinder outperforms the traditional peak finding method SNAP arises when the fragment composition substantially differs from that of the averagine used. As shown in Figure 2·1, selecting an appropriate averagine that fits all GAG fragments is difficult due to the variable number of sulfur atoms in the fragments. Compound #9 contains a heavily sulfated reducing end, with three sulfate groups on one GlcNAc. While GAGfinder finds the Y1-S and Y1 ions for this compound and scores them in the top ten, SNAP is unable to find them. Figure B·5 shows the annotated spectra, using the top 20 (or so) most intense fragments for each saccharide, and Figure B·6 shows the portion of the spectrum containing these fragments. In both cases, there are other isotopic distributions interspersed, but none of these precisely overlap with their peaks.

Wolff and colleagues first showed how metal cationization can help curb sulfate loss in EDD [103], an approach that has gained popularity in the years since. While our group typically avoids metal adduction during GAG analysis due to the negative effects on the instrument and the extra work up, we nonetheless designed GAGfinder to be able to handle samples that have been cationized. In GAGfinder, cationization

adds to the search space, and therefore the runtime, without necessarily improving peak finding performance, reduced sulfate loss aside. While metal cationization can help remove the ambiguity of tandem mass spectra, allowing for easier GAG sequencing, its utility is seen mostly in that step of the sequencing pipeline. GAGfinder is only looking for fragments and isotopic distributions of given compositions, regardless of whether there is metal cationization or not, and therefore, metal cationization should not affect GAGfinder's peak finding performance.

In conclusion, use of GAGfinder will allow researchers to swiftly and accurately assign elemental compositions and product ion types to product ions in GAG saccharide tandem mass spectra. While GAGfinder was tested exclusively on pure, synthetic compounds, we are evaluating its ability to assign product ion m/z, charge state, and elemental composition for biological samples. Finally, we demonstrate that the use of a brute force method for peak finding balances search space size and overall analysis time compared to traditional methods.

Chapter 3

GAGrank: Software for Glycosaminoglycan Sequence Ranking Using a Bipartite Graph Model

3.1 Introduction

The sulfated glycosaminoglycans (GAGs) are long, linear polysaccharides that can be found as the glycan portion of proteoglycans (PGs) on cell surfaces and in extracellular matrices (ECMs). There are three classes of sulfated GAG, each with its own distinct repeating disaccharide unit (Figure 1·1) and biology. Heparan sulfate (HS) has been shown to participate in or affect blood coagulation [104], growth factor signaling [105], angiogenesis [106], and cell proliferation and migration [107]. Chondroitin sulfate (CS) has been shown to participate in or affect brain development [108], spinal cord injury and neuroregeneration [109], neural stem cell migration [110], and osteoarthritis [111]. Keratan sulfate (KS) has been shown to participate in or affect corneal hydration [112], infection and wound repair [113], and cell migration [114]. As a part of membrane PGs and the ECM, GAGs bind numerous growth factors and

growth factor receptors and thereby mediate cell-cell, cell-matrix, and host-pathogen interactions. As such, the ability to sequence GAGs quickly and accurately is an important step in understanding how changes in GAG sequences impact biological mechanisms.

Algorithms based on THRASH [90] for identification of monoisotopic peaks and estimation of elemental compositions do not suffice for GAGs due to the fact that the sulfur and oxygen contents vary significantly among fragment ions, thus precluding use of a single averagine for elemental composition approximation. Our group recently developed a GAG-specific algorithm performing both of these steps, GAGfinder (see Chapter 2). GAGfinder provides a list of peaks and annotations from a MS^2 experiment for the sequencing pipeline.

In a typical MS^2 experiment, spectra are pre-processed depending on the type of mass analyzer used. For ion cyclotron resonance and Orbitrap analyzers, the signal is transformed from the time domain to the m/z domain, thereby filtering background noise. The resulting m/z domain spectra can be interpreted using mass spectrometry software.

Electron activated dissociation (ExD) is a general term that refers to use of ion-ion or ion-electron reactions to dissociate the analyte. For fragmentation of anionic species, including GAGs, ExD includes electron detachment dissociation (EDD) [115], where the analyte is fragmented by detaching an electron with an electron beam, and negative electron transfer dissociation (NETD) [116], where the analyte is fragmented by transferring an electron from the anionic species to a cationic reagent. Wolff and colleagues first demonstrated the efficacy of ExD for dissociating GAG oligosaccharides in various applications, using both EDD [80, 117, 118] and NETD [88]. Huang and colleagues have also shown the utility of ExD for GAG oligosaccharides in terms of reducing labile sulfate loss [89]. Clearly, ExD methods show promise as the analytical

tool of choice in GAG sequencing.

There are numerous existing methods for computational GAG sequencing, which are described in detail in Chapter 1.4. These methods have succeeded in making GAG sequencing faster and easier, but they each have drawbacks. HOST [72], the GAG update to Glycoworkbench [73], and the approach described by Spencer *et al.* [75] use the results of disaccharide analysis to guide their algorithm, meaning that the methods are inappropriate for top- or middle-down glycomics studies. HS-SEQ [76] shows promise in locating site-specific sulfation for HS oligosaccharides, but requires prior knowledge about the HS backbone and does not handle mixtures as would be seen in an LC-MS^2 experiment. Furthermore, it only considers HS oligosaccharides, and does not work on CS or KS GAGs. GAG-ID [77] shows promise in ranking individual GAG sequences mapping to a given GAG composition, but requires an extensive chemical workup involving permethylation, desulfation, and pertrideuteroacetylation. Duan and Amster's genetic algorithm [79] shows great promise for reducing search space and computation time, but is a non-deterministic algorithm and therefore cannot guarantee to reach a global optimum. We sought to develop a novel, deterministic GAG sequencing method that has fewer steps before use than existing methods but still delivers optimal performance.

At the core of any sequencing method using MS^2 data is the relationship between the unknown sequence and its fragments: the actual sequence is ascertained based on the fragment ions generated in the fragmentation process. For GAGs, there are often many possible sequences for a given composition, and in an ExD experiment, there is a rich complement of product ions in the spectrum. The relationship between possible sequences and observed product ions is many-to-many, and can be represented in a network structure. In particular, the network structure is that of a bipartite network, which is a network whose nodes can be separated into two distinct partitions with

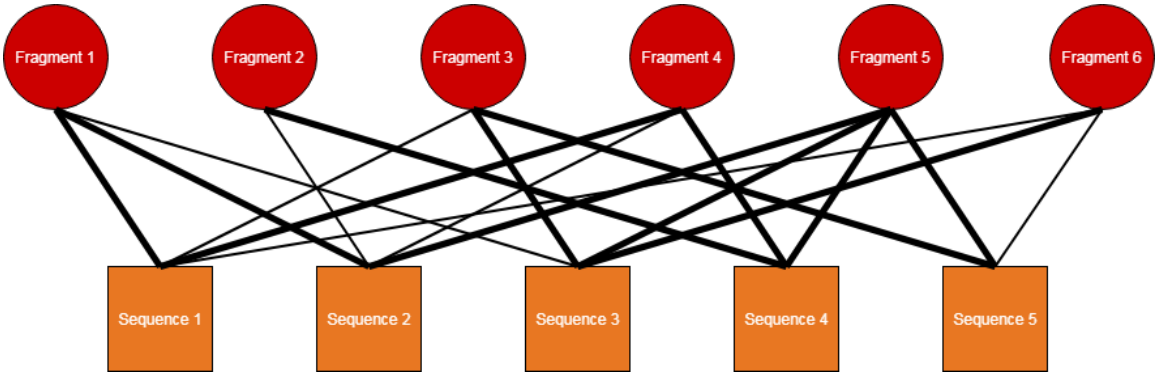


Figure 3·1: Example bipartite network of sequences and fragments. This is a toy visualization of a bipartite network of sequences and fragments. In this case, there are five sequences and six fragments. An edge between a sequence and a fragment denotes that fragment being a possible fragment for that sequence. The edge width denotes the type of fragment for that sequence; a wider edge represents a terminal fragment, while a narrower edge represents an internal fragment.

edges only connecting nodes in one partition to nodes with the other partition. Figure 3·1 shows a graphical representation of the bipartite network relationship between potential sequences and product ions.

The determination of node importance has been a topic of significant interest in network analysis, in particular for social networks [119], protein-protein interaction networks [120], and the World Wide Web [121], among many others. The concept of centrality in network analysis aims to solve this problem, and there are numerous existing algorithms for computing centrality measures. One such method is PageRank [122], developed by Brin and Page in 1996 for Google as a way to rank webpages according to their importance for search engine optimization purposes. Briefly, PageRank gives webpages higher importance values if they are linked to by other important webpages. PageRank was developed for general networks (i.e. not bipartite networks), but a recent method, BiRank [123], was developed that adapts the PageRank algorithm for the specific case of bipartite networks. The algorithm pseudocode for BiRank is shown in Algorithm 3.1. Briefly, BiRank gives nodes in

partition A higher importance if they are linked to important nodes in partition B, and vice versa. Because of its design for bipartite networks, we employed BiRank with the goal of determining precursor sequence based on fragmentation patterns in the first GAG sequencing method developed using a network structure and network analysis algorithm, GAGrank. GAGrank was developed as a command line interface in the Python language. This paper describes the method, and demonstrates its performance on a set of GAG standards.

Algorithm 3.1: BiRank Algorithm, adapted from [123]

Input: Weight matrix W, query vectors $\mathbf{p}^0$, $\mathbf{q}^0$, and hyper-parameters α, β;
Output: Ranking vectors $\mathbf{p}$ and $\mathbf{u}$;

Symmetrically normalize W: $S = D_u^{-\frac{1}{2}} W D_p^{-\frac{1}{2}}$;
Randomly initialize $\mathbf{p}$ and $\mathbf{u}$;
while *Stopping criteria is not met* **do**
    $\mathbf{p}^i \leftarrow \alpha S^T \mathbf{u}^{i-1} + (1-\alpha)\mathbf{p}^0$;
    $\mathbf{u}^i \leftarrow \beta S \mathbf{p}^{i-1} + (1-\beta)\mathbf{u}^0$;
    $i \leftarrow i + 1$;
return $\mathbf{p}$ *and* $\mathbf{u}$

3.2 Experimental Procedures

3.2.1 GAGrank Overview

Figure 3·2 shows the steps in the GAGrank algorithm, the details of which are presented in the next several subsections.

Inputs – GAGrank has several inputs, both required and optional. The required inputs are the peak/fragment list, the GAG class, the precursor ion m/z, and the precursor ion charge, assuming unadducted deprotonated ions. The peak/fragment list must be an output data file from our previous work, GAGfinder (see Chapter 2), and the GAG class being analyzed must be denoted by its initials (i.e. HS, CS, or KS). The optional inputs are the reducing end tag and the number of sulfate losses to

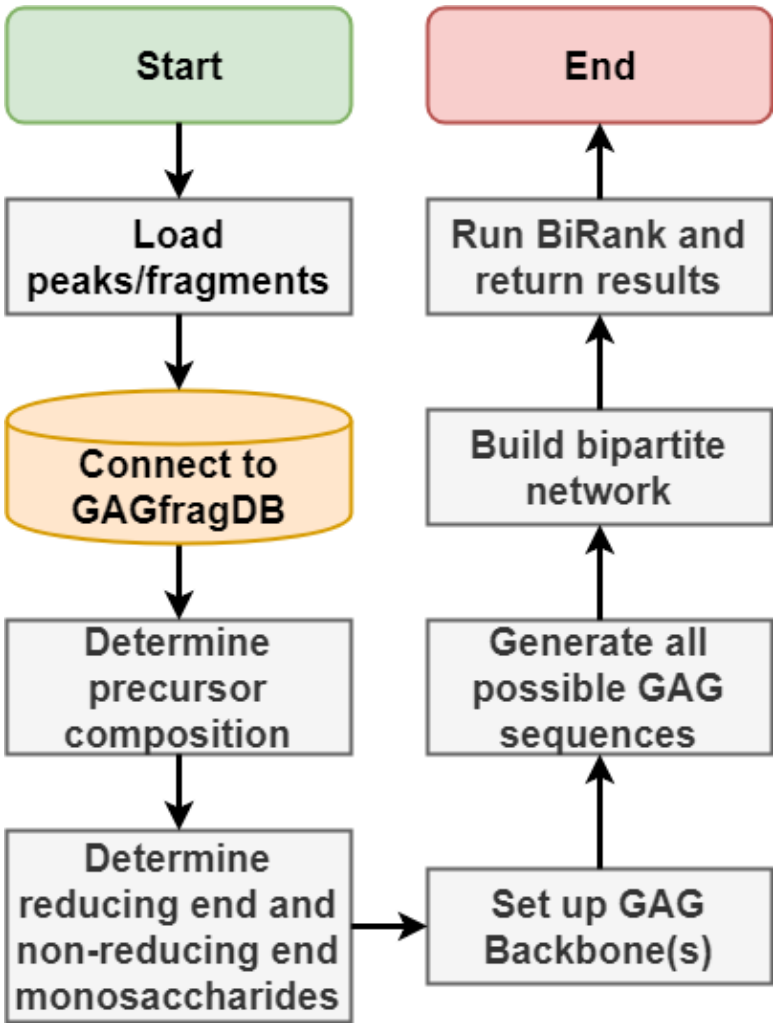


Figure 3·2: Workflow for GAGrank algorithm. The steps in GAGrank's algorithm.

consider. If the analyte was tagged on the reducing end to break potential molecular symmetry, the user must denote what elements the tag adds to the sequence. For instance, if the reducing end tag is 4-nitrophenol (PNP), the tag should be encoded as $C_6H_3NO_2$ rather than the PNP elemental composition of $C_6H_5NO_3$, since that is the number of each element that is added to the structure. For deciding an appropriate number of sulfate losses for GAGrank to use, we recommend using the floor of two times the free proton index (FPI) described in [124]. However, this input is up to the user's discretion.

Step 1: Load peak/fragment list – First, GAGrank loads the peak/fragment list returned by GAGfinder into Python as a NumPy array with two columns, fragment and G score. The G score is GAGfinder's goodness-of-fit score for fitting experimental isotopic distributions found in spectra to theoretical isotopic distributions. A smaller G score represents a better fit between the two distributions.

Step 2: Determine precursor composition – Next, GAGrank utilizes the database GAGfragDB to determine the precursor composition in a manner similar to GAGfinder. GAGfragDB was developed in SQLite to store every possible fragment for a given precursor composition, but it also stores useful information about precursors, such as their chemical formula and monoisotopic mass. GAGrank selects the precursor composition by comparing the neutral mass of the spectral precursor ion to the list of neutral masses in GAGfragDB and picking the one that is arithmetically closest.

Step 3: Determine reducing end and non-reducing end monosaccharides – The next step in GAGrank's pipeline is also similar to one found in GAGfinder. By evaluating the number and type of monosaccharides present in the precursor's composition, we can potentially determine the order of the monosaccharides in the oligosaccharide backbone. For a detailed description of this process, see Chapter 2.

Step 4: Set up GAG backbone(s) – We can build the backbone(s) of the GAG

sequence using our understanding of GAG sequence construction and the terminal sugar residues determined in step 3. In the event that we cannot determine the terminal sugar residues, we must consider two backbones; one with amino sugars in the odd positions in the backbone, and another with amino sugars in the even positions in the backbone. Given the backbone(s) of sugar residues and the GAG class for the structure, we can define the positions for potential modifications (Figure 1·1).

Step 5: Generate all possible GAG sequences – We now have the backbone(s) of the GAG, the potential modification positions along the backbone(s), and the number of each modification (sulfation and acetylation). We use combinatorics to generate each possible sequence for a given composition.

Step 6: Build bipartite network – Using Python's NetworkX module [125], we encode the relationships between each potential sequence and each fragment found by GAGfinder in a bipartite network. For each potential sequence, we derive its potential fragments by generating all terminal glycosidic fragments, terminal cross-ring fragments, and internal glycosidic-glycosidic fragments. We do not consider internal glycosidic-cross-ring or internal cross-ring-cross-ring fragments because they are rare and of low-abundance, and do not add much additional information about the sequence. We then compare this list of potential fragments to the those found in the spectrum loaded in step 1, and place edges between the sequence and each fragment in the intersection. Equation 3.1 shows how we encode the edge width for these edges. The values for Equation 3.1 are based on those used in our previous work, HS-SEQ [76]. In cases where a fragment could be both a terminal fragment and an internal glycosidic-glycosidic fragment, the edge width is selected as a terminal fragment. The tuning parameter *r1* controls the effect that double glycosidic bond fragments has on the performance of BiRank.

$$w_{xy} = \begin{cases} 1.0, & \text{if fragment } x \text{ is a terminal fragment in sequence } y \\ 0.2^{r1}, & \text{if fragment } x \text{ is a double glycosidic fragment in sequence } y \end{cases} \tag{3.1}$$

Step 7: Run BiRank and Return Results – The final step in GAGrank's pipeline is to run the BiRank algorithm [123] on the network built in step 6. The inputs for BiRank include the graph's weight matrix W, query vectors $\mathbf{p}^0$ and $\mathbf{u}^0$, and hyper-parameters α and β. The weight matrix is symmetric, consisting of the edge weights between nodes in the graph, as described in Equation 3.1. For nodes with no edge between them, the weight w_{xy} is given as 0. The query vectors store a prior belief about the ranking criterion for the sequences and fragments before iterating through the BiRank algorithm. For our purposes, we consider $\mathbf{p}$ to be the fragments vector and $\mathbf{u}$ to be the sequences vector. The fragments' query vector values are calculated using equation 3.2:

$$p_x^0 = \frac{I_x}{G_x^{r2}} \tag{3.2}$$

For fragment x, we assign the query value as its intensity divided by its GAGfinder G score. The tuning parameter *r2* controls the effect the G score has on the overall score. The sequences' query vector values are calculated using Equation 3.3:

$$u_y^0 = (\prod_m score_m)^{r3} \tag{3.3}$$

For sequence y, we assign the query value as the product of the residue likelihood scores for each monosaccharide residue in the sequence. The residue likelihood is calculated using Equation 3.4:

$$
\begin{aligned}
score_m &= 1.0 - 0.6N_m - 0.3S_m \\
N_m &= \begin{cases} 1, & \text{if amine is unoccupied} \\ 0, & \text{otherwise} \end{cases} \\
S_m &= \begin{cases} 1, & \text{if 3-}O\text{-sulfation without 6-}O\text{-sulfation} \\ 0, & \text{otherwise} \end{cases}
\end{aligned} \tag{3.4}
$$

Each residue has a maximum likelihood value of 1.0. If the residue is an amino sugar that has a free amine group, the value is decreased by 0.6. If the residue is an HS GlcN residue that is 3-O-sulfated and not also 6-O-sulfated, the value is decreased by 0.3. These deductions account for the rarity of free amines and 3-O-sulfation without 6-O-sulfation in nature. The tuning parameter *r3* controls how much a sequence with rare modification patterns is punished prior to running the BiRank algorithm. The hyper-parameters α and β control how much of each iteration's ranking score is due to the query vectors for the fragments and sequences, respectively. A larger value for either hyper-parameter weights the iterating results of BiRank more than the query vector. Once the BiRank algorithm iterates to convergence, GAGrank outputs the ranking of sequences with their ranking score into a tab-delimited file. A larger score represents a higher ranking.

3.2.2 Data Acquisition and Preprocessing

We selected thirteen pure synthetic GAG standards on which to train and validate GAGrank and two isomeric mixtures of pure synthetic GAG standards to show GAGrank's performance on mixtures, shown in Figure 3·3. These samples were selected for their varying lengths, modification amounts and patterns, disaccharide order, and precursor charge. Ten pure synthetic standards were selected as train-

ing data and three pure standards were selected as validation data. Training compounds 1, 3, 5, 8, 9, and 10 were acquired through a publicly available set of HS standard saccharides funded by the NIH and maintained by the Zaia laboratory (http://www.bumc.bu.edu/zaia/gag-synthetic-saccharides-available/). The remaining training compounds, all of the validation compounds, and the compounds mixed in the isomeric mixtures were synthesized as described [99, 126–128]. Each of the two mixtures was tested in ratios of 100:0, 90:10, 70:30, 50:50, 30:70, 10:90, and 0:100.

Each sample was subjected to either electron detachment dissociation (EDD) or negative electron transfer dissociation (NETD) using a Bruker SolariX 12T FTMS instrument. The spectra were converted to centroided mzML using the compassXport command line utility 3.0.13 (Bruker Daltonics, Inc.). Elemental compositions of tandem mass spectral peaks corresponding to GAG fragments were determined using a modified version of GAGfinder (see Chapter 2) that requires an isotopic distribution to have peaks A and A+1 to have an intensity above the noise threshold, with error tolerance between 3 and 5 ppm and considered sulfate losses determined by the floor of two times the FPI.

3.2.3 Parameter Optimization

In order to determine optimal values for the above parameters, we employed the simulated annealing (SA) probabilistic optimization procedure. SA is named after the process of annealing in metallurgy whereby material is heated to the point where its geometric structure breaks down and it can be shaped, followed by a slow cooling to re-establish the geometric structure. SA works by randomly moving from one solution to a neighboring solution until a very good, though not necessarily perfect, solution is found. During the course of the SA algorithm, if the new solution has a better fitness than the current solution, the new solution will always be selected; however, if the new solution has a worse fitness than the current solution, the new

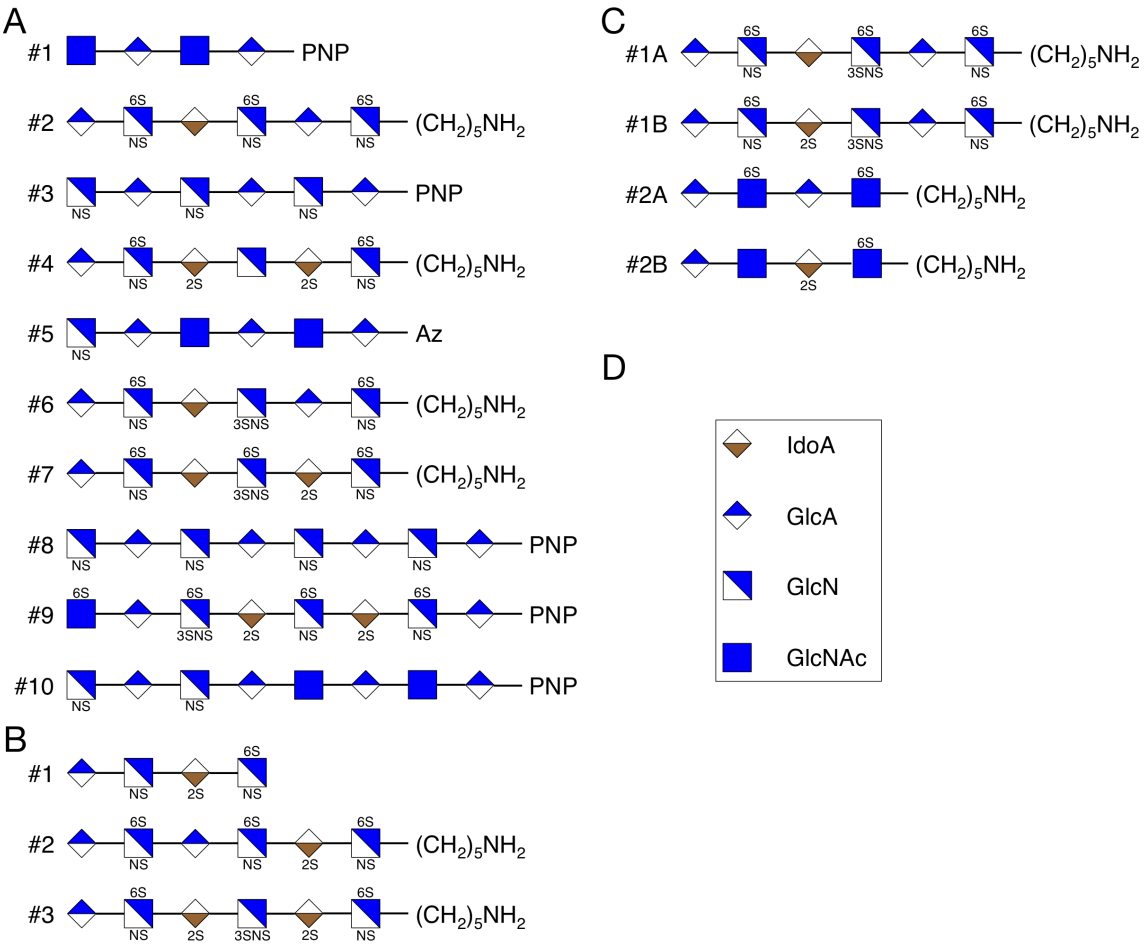


Figure 3·3: Structures analyzed in this study. A. The ten training saccharides. B. The three validation saccharides. C. The two isomeric mixtures. D. Key for the symbols in the figure. Each analyzed structure was dissociated via NETD, except testing compounds #6 and #7, which were dissociated via EDD. The precursor charge states for these compounds range from -2 to -6. The compounds were selected to represent diversity in chain length, modification amounts and patterns, and charge state for GAG oligosaccharides.

solution may be selected based on a probabilistic criterion. Equation 3.5 shows the formula for calculating the probability of moving to a worse solution:

$$P(move) = e^{\frac{f_{new} - f_{current}}{T}} \tag{3.5}$$

Here, f_{new} and $f_{current}$ represent the fitness scores for the new solution and the current solution, respectively. T represents a "temperature" parameter that is analogous to the cooling process in annealing, described above, and "cools" from a value of 1 to 0. For cases where the new solution's fitness is worse than the current solution's, when T is close to 1, the algorithm will probably still move to that solution, while when T is close to 0, the algorithm will probably stay at the current solution. Therefore, at the beginning of the algorithm, it is more likely to move to a worse solution than toward the end. This helps combat the problem of local maxima.

In our implementation, we employed SA to find a very good solution for GAGrank on our ten training oligosaccharides. We optimized the five aforementioned parameters as well as the number of fragments returned by GAGfinder. To identify a new solution we randomly selected one of the six parameters and randomly changed its value. For α and β, the value could be any number between 0 and 1. For *r1*, *r2*, and *r3*, the value could be any number between 0 and 10. For the number of fragments used, the value could be any integer between 5 and 100, or all of the found fragments. We rounded the value for α and β to the hundredth decimal point and we rounded the value for *r1*, *r2*, and *r3* to the tenth decimal point. We reduced T by multiplying it by 0.9. In order to slow the SA process and more completely explore the search space, we remained at each value of T for 100 iterations. We calculated the fitness of each solution as the average percent of incorrect sequences with a worse BiRank score than the correct sequence. For example, if a composition has ten possible sequences, the solution's fitness is equal to 1.0 (9/9) if the correct sequence has the highest BiRank

score, 0.0 (0/9) if the correct sequence has the lowest BiRank score, and 0.778 (7/9) if the correct sequence has the third-highest BiRank score.

3.3 Results

3.3.1 Optimal Parameters

Parameter optimization via simulated annealing found multiple combinations of parameters that resulted in optimal performance on the training data, which are summarized in Table 3.1. These combinations all returned a fitness value across the ten training compounds of 0.9997, meaning that, on average, GAGrank returned a better ranking score for the correct sequence than 99.97% percent of incorrect sequences. Each parameter combination follows similar patterns: large values for *r1* and *r2*, small values for *r3*, large values for α and β, and between 61 and 97 GAGfinder fragments used. The large values for *r1* can be interpreted as evidence that internal double glycosidic bond fragments are far less important for GAG sequencing than terminal fragments. The large values for *r2* can be interpreted as evidence that the fragments' goodness-of-fit G scores from GAGfinder are more important factors in GAG sequencing than their intensities. The small values for *r3* can be interpreted as evidence that rare modifications do not need to be punished severely for sequences without rare modifications to perform well. The large values for α and β can be interpreted as evidence that the initial ranking scores for the fragments and sequences are much less important to the optimal performance than the placement and widths of the edges in the graph structure. Finally, the range for the number of fragments to input into GAGrank mostly relates to the number of fragments initially found by GAGfinder; for some saccharides, GAGfinder found fewer than 60 fragments in the spectrum, while for others, GAGfinder found well over 100. The range of 60-70 suffices to get positional detail for modifications without introducing false positives.

Table 3.1: Summary statistics for each GAGrank parameter that resulted in the best performance on the testing compounds.

	r1	*r2*	*r3*	α	β	**# fragments**
Minimum	0.5	0.9	0.1	0.77	0.76	61
Maximum	9.8	9.6	1.7	0.99	1.00	97
Mean	5.5	5.2	0.6	0.93	0.92	70
Median	5.4	5.1	0.4	0.98	0.94	68
Mode	9.3	4.9	0.1	0.98	0.94	64

Table 3.2 shows the overall ranking and percent of incorrect sequences outscored for each of the training compounds in GAGrank and Tables C.2-C.11 show the top ten (or fewer) GAGrank outputs for each. For eight of the ten training compounds, GAGrank returned the correct sequence with the best ranking score out of all of the possible sequences. In test compound #6, GAGrank returned the correct sequence tied with two other sequences for the second-best ranking score out of all of the possible sequences. This is likely due to the effect that test compound #6's rare 3-*O*-sulfation without 6-*O*-sulfation has on the prior sequence rankings; indeed, Table C.7 shows the top four sequences all differ only by the presence (or absence) and location of the 3-*O*-sulfation in the sequence. In test compound #9, GAGrank returned the correct sequence tied with 25 other sequences for the best-ranking score out of all of the possible sequences, as seen in Table C.10. This is likely due to the large number of possible sequences for that particular composition, combined with the relative dearth of fragments found by GAGfinder.

3.3.2 Parameter Validation

Table 3.3 shows the ranking score, overall ranking, and percentile of each of the validation compounds in GAGrank and Tables C.12-C.14 show the GAGrank outputs for each. Similar to the results for the test compounds, GAGrank returned the correct sequence with the best ranking score out of all of the possible sequences, while GAGrank

Table 3.2: GAGrank performance for the test compounds using any of the optimal parameter combinations.

Test Compound	Ranking	% incorrect outscored
#1	#1 out of 2	100%
#2	#1 out of 1,848	100%
#3	#1 out of 440	100%
#4	#1 out of 1,584	100%
#5	#1 out of 60	100%
#6	#2-#4 out of 1,848	99.8%
#7	#1 out of 990	100%
#8	#1 out of 3,640	100%
#9	#1-#26 out of 23,298	99.9%
#10	#1 out of 1,092	100%

returned the correct sequence tied with one other sequences for the third-best ranking score out of all of the possible sequences for the compound with a single 3-*O*-sulfation without a 6-*O*-sulfation. As can be seen in Table C.14, for validation compound #3, the two sequences with a better ranking score than the actual sequence did not have any rare modifications, while the actual sequence and the incorrect sequence with which it tied both have one residue with 3-*O*-sulfation without 6-*O*-sulfation. Unlike the results for test compound #6, one of the two sequences with a better ranking score than validation compound #3 did not have the correct modification numbers at each residue in the sequence; the sequence that had the second-best ranking score placed a sulfate at the 2-*O* position of the non-reducing end GlcA rather than at the 6-*O* position of the neighboring GlcN.

Table 3.3: GAGrank performance for the test compounds using any of the optimal parameter combinations.

Validation Compound	Ranking	% incorrect outscored
#1	#1 out of 140	100%
#2	#1 out of 1,584	100%
#3	#3-#4 out of 990	99.7%

3.3.3 GAGrank and GAG Mixtures

Figure 3·3C shows the structures of two pairs of saccharide isomers used to show the ability of GAGrank to analyze mixtures. Table 3.4 shows the rankings for each compound in each of the two mixtures at each of the ratios and Tables C.15-C.28 show the GAGrank outputs for each. As in the test compounds and validation compounds, one of the sequences, mixture compound #1B, has a rare modification, 3-*O*-sulfation without 6-*O*-sulfation. Further, this sequence never has the best ranking score at any mixture ratio, just as in the similar cases in the test compounds and validation compounds. The sequences corresponding to the remaining three compounds have the highest-ranking score when they comprise at least 70% of the isomeric mixture of which they are a part. Furthermore, each of the compounds used in the mixtures performs as well as it does when it is pure as long as it is 70% or more of the isomeric mixture.

3.3.4 Runtime Analysis

Information about the runtime of GAGrank on each of the compounds and mixtures is available in Supplemental Table C.1. With the exception of validation compound #9, whose composition has 23,298 different possible sequences, GAGrank ran to completion in under 17 seconds for each compound, with many running to completion in under 10 seconds. There is a strong relationship between the number of possible sequences for a compound's composition and the runtime. GAGrank was tested on a 2011 MacBook Pro that has a 2.4 GHz Intel Core i5 processor with 4 GB RAM. GAGrank should run even faster on a more modern machine with greater computational resources.

Table 3.4: GAGrank performance for the mixture compounds using any of the optimal parameter combinations.

Mixture and ratio	Compound A rank	Compound A % incorrect outscored	Compound B rank	Compound B % incorrect outscored
Mixture #1 100:0	#1 out of 1,584	100%	–	–
Mixture #1 90:10	#1 out of 1,584	100%	#10-#11 out of 1,584	99.4%
Mixture #1 70:30	#1 out of 1,584	100%	#8-#9 out of 1,584	99.6%
Mixture #1 50:50	#22 out of 1,584	98.7%	#2-#3 out of 1,584	99.9%
Mixture #1 30:70	#64 out of 1,584	96.0%	#2-#4 out of 1,584	99.8%
Mixture #1 10:90	#109 out of 1,584	93.2%	#2-#4 out of 1,584	99.8%
Mixture #1 0:100	–	–	#2-#4 out of 1,584	99.8%
Mixture #2 100:0	#1 out of 30	100%	–	–
Mixture #2 90:10	#1 out of 30	100%	#6 out of 30	85.7%
Mixture #2 70:30	#1 out of 30	100%	#5 out of 30	89.3%
Mixture #2 50:50	#2 out of 30	100%	#1 out of 30	100%
Mixture #2 30:70	#3 out of 30	96.4%	#1 out of 30	100%
Mixture #2 10:90	#4 out of 30	92.9%	#1 out of 30	100%
Mixture #2 0:100	–	–	#1 out of 30	100%

3.4 Discussion

Here, we have presented our work on bipartite network representations and analyses for the relationship between GAG sequences and MS^2 fragment ions, GAGrank. GAGrank is an algorithm that ranks nodes using the bipartite network's structure and prior information about the sequences and fragments, giving each node an importance score. GAGrank is currently available in command line form. We plan to merge it and our previous work, GAGfinder (see Chapter 2), into a GAG sequencing pipeline with a graphical user interface in the near future. The command line interface is easy to use, with only a few arguments required for operation.

To our knowledge, this is the first time this approach has been used for the problem of GAG sequencing, and it has certain inherent advantages. One such advantage is that the concept of a relationship between sequences and fragments is intuitive and easy to visualize. Another advantage is that bipartite networks have been exhaustively studied in other fields, meaning that methods for analyzing them have already been developed. GAGrank, at its most basic level, is simply an implementation of one of these methods, BiRank [123]. Furthermore, enumerating every sequence that is possible for a given GAG composition allows for ranking sequences by their importance in the network, which is analogous to their likelihood.

We used three separate sets of GAG compounds for training and validation. We optimized GAGrank's parameters using the ten compounds in our training set, and found numerous sets of parameters that returned a near-optimal solution. Using these parameters, we tested GAGrank's performance on the three compounds in our validation set, and GAGrank returned a similarly near-optimal solution for these compounds. We also tested GAGrank's ability to sequence GAG mixtures on two separate isomeric mixtures that differed only in one positional sulfation. On these mixtures, GAGrank performed well, ranking the sequence that made up more of the mixture highly while ranking the sequence that made up less of the mixture lower. An intuitive way to view GAGrank's performance on mixtures is that, the higher percentage of the mixture a particular sequence is, the higher that sequence ranks. While GAGrank's performance on mixtures shows that this method has potential for characterizing mixture constituents, there is currently no means by which users can determine that their sample is a mixture.

For the cases in the test set, validation set, and mixture set where the actual sequence did not rank highest of all the possible sequences, each compound had a rare modification (3-*O*-sulfation without 6-*O*-sulfation on a glucosamine residue)

that was penalized in the sequences' query vector. A simple solution to this problem would be to not punish sequences with rare modifications, but we hypothesize that this would penalize the final performance of sequences that are much more common in nature. In the course of parameter optimization, an α equal to 1.00 was tested numerous times, but never returned the best solution. This case (α=1.00) means that the sequence ranking is derived entirely from the graph structure, without any input from the query vectors. Without a near-full complement of fragments in the spectrum, there will be many sequences that have the exact same edges, and without prior information, GAGrank cannot distinguish them. We believe that the benefit of teasing out the exact correct sequence when it has no rare modifications outweighs the slightly worse performance for those sequences that do have a rare modification.

There are a couple of downsides for GAGrank, both of which are mostly about user preference. The first is that it requires a peak list from GAGfinder that contain correctly fit elemental compositions, and will not work on peak lists exported from the vendor MS^2 software generated using averagine approximations. While this adds an extra step into the pipeline that other programs may not have, it uses the most appropriate means of assigning monoisotopic peaks and elemental compositions. We have demonstrated the efficacy and speed of GAGfinder in that project's manuscript (see Chapter 2). Another downside is that GAGrank was not developed to work on metal cationized compounds. Wolff and colleagues were the first group to show how metal cationization reduces sulfate loss for EDD-dissociated HS compounds [103], and this approach succeeds in this endeavor. However, including saccharide ions that have been cationized can severely increase the search space, making the sequencing problem intractable. Furthermore, none of the samples in this manuscript were cationized, and GAGrank performed well even with the higher amounts of sulfate loss.

Of course, GAGrank was tested on pure synthetic saccharides, but biological data

is typically noisy and not pure. A typical experiment that generates biological GAG data uses liquid chromatography-tandem mass spectrometry (LC-MS^2). In LC-MS^2, samples can be separated in the LC column by a number of different means, including charge or size, and an online mass spectrometer generates MS^2 spectra as samples elute off of the column. This results in a large number of spectra that contain mixtures of GAG structures. We demonstrated GAGrank's performance on mixtures of pure chemicals, and showed that there is potential there, but GAGrank is not yet ready to handle such large amounts of high throughput data, and it does not perform as well on mixtures as it does on pure samples. We will continue to develop GAGrank to handle these situations.

In conclusion, GAGrank demonstrates excellent performance in the difficult task of GAG sequencing. It ranks sequences accurately based on the complement of fragments found via GAGfinder, and will be a valuable resource for GAG researchers who need fine structure detail for their samples.

Chapter 4

Discussion and Future Work

4.1 Summary of Dissertation

The original work described in this dissertation falls into the realm of GAG sequencing and is intended to be two parts of a modular GAG sequencing pipeline. Any GAG sequencing project will include four steps: sample preparation/extraction, LC-MS^2, peak determination and annotation, and sequence identification. GAGfinder and GAGrank are valuable software tools for helping speed up the final two steps in the pipeline. However, due to the large number of spectra associated with an LC-MS^2 experiment, parallelization techniques will need to be employed to ensure an acceptable runtime. The use of a high-performance computing cluster would be necessary to attain this goal.

Chapter 1 is an overview and review of automated sequencing methods for all classes of glycans. It offers background biological information on glycans, a brief overview of MS^2 techniques, a review of existing sequencing methods for branched glycans and GAGs, and a critical view on GAG sequencing. The methods described in Chapters 1.3 and 1.4 were inspired by previous work in proteomics and will inspire future techniques in the pursuit of a gold standard sequencing approach for glycans.

Chapter 2 describes GAGfinder, the first GAG-specific peak finding software. Traditional peak finding methods that were designed for proteins and rely on averagine are improper for GAGs due to their non-uniform distribution of sulfation and the relatively intense isotopic A+2 peak of ^{34}S. The brute force method considers all possible fragments for a GAG composition and compares the theoretical isotopic distribution to the spectral data. I showed GAGfinder's performance on a set of ten synthetic saccharides and it performs mostly in line with vendor peak finding software, with the added bonus of automated annotation. Despite being a brute force method written in Python, GAGfinder has an acceptable runtime, allowing for rapid analysis of GAG spectra. In order to handle the large number of spectra associated with high throughput online LC-MS^2, parallelization methods will need to be developed to handle the large number of spectra and a more powerful programming language (e.g. C or C++) will need to be employed.

Chapter 3 describes GAGrank, a GAG sequencing method that ranks GAG structures in order of likelihood based on the output of GAGfinder. GAGrank configures the possible structures and found fragments into a bipartite graph, and uses a node ranking approach based on a bipartite extension, BiRank [123], of Google's PageRank algorithm [122] for ranking search results. I used simulated annealing to optimize the BiRank parameters using ten synthetic training saccharides, and confirmed the performance using three synthetic validation saccharides. I also showed how GAGrank performs on two isomeric mixtures of GAG samples, since many GAG sequencing projects are LC-MS^2 analyses of biological samples, which sort along the LC column by their interactions with the mobile and stationary phases.

The remainder of this chapter contains a general discussion of each software's role in the GAG sequencing pipeline, a description of another method for GAG sequencing that met with poor results, and future directions for these projects. There are

three appendices after this chapter. Appendix A.1 contains a short description of a web application for proteomics statistical analysis. Appendix A.2 and appendix A.3 contain supplemental material for Chapters 2 and 3, respectively.

4.2 General Discussion

The first algorithm for biomolecule sequencing was BLAST [129], developed in 1990, and could be considered a precursor to the expansion of bioinformatics in the human genome project; truly, sequencing has been a major interest in bioinformatics since its beginning. Sequencing methods for biomolecules such as DNA and RNA are established and have the advantage of sample amplification using polymerase chain reaction (PCR). Sequencing methods for proteins are equally established, and have the advantage of the use of a reference genome. Sequencing methods for GAGs do not have these advantages, and therefore require a more nuanced approach. Furthermore, GAG sequencing is tricky, due to their non-uniform distribution of sulfate modifications, the epimerization of C5 on uronic acid residues, and facile sulfate losses during MS^2. This dissertation has described work I have performed in the pursuit of a standardized and effective GAG sequencing pipeline.

GAGfinder was developed for more accurate isotopic cluster finding for GAG MS^2 spectra and peak annotation. The conventional method by which MS^2 spectra clusters are retrieved is the THRASH algorithm [90]. This method works by fitting potential isotopic clusters to the average molecular building block, or averagine, for the class of molecule being tested. This approach works well for protein MS^2 analysis, since the isotopes in peptide fragments typically behave similarly. Where this method runs into trouble is cases where the averagine does a poor job of modeling the isotopic pattern of the fragment. These would be cases where the elemental composition changes dramatically along the molecule, as is the case for GAGs. The same structure could

have every possible modification location occupied on one end and unoccupied on the other end, which would lead to entirely different isotopic distributions for fragments on each end, as demonstrated in Figure 2·1.

The advantage that averagine-reliant cluster finding methods have is speed, compared to enumerating each possible combination of subunits. Methods such as Bruker Daltonics' SNAP algorithm only make a single pass through the spectrum, comparing any potential isotopic clusters to averagine as they are found. GAGfinder utilizes a brute force method that enumerates and tests all possible fragments at all possible charge states. This approach works for GAGs due to the relatively limited number of possible structures for a given chemical composition, as shown in Figure 2·2, and the runtime numbers shown in Table 2.3 show that speed is not an issue.

Figure 4·1 shows the distributions of hits and misses for all fragments tested by GAGfinder for sequence #6. The distribution for the hits is to the left of that for the misses, meaning that their G scores are lower overall than the misses'. This shows that GAGfinder discriminates between hits and misses properly, and mimics the behavior of the performance test described in Chapter 2.2.3. A random sampling of hits and misses would not have two separate G score distributions, as is seen in the figure.

GAGrank was developed as a technique for ranking GAG sequences by likelihood based on the isotopic clusters observed in a MS^2 spectrum. There are numerous existing methods for this task, each with its own advantages and drawbacks. GAGrank takes into consideration the reality that sulfate loss happens during the course of GAG MS^2, does not require any additional chemical workup, and is deterministic. Its bipartite network structure represents a natural depiction of the relationship between possible structures and found fragments, and allows for established network-based ranking algorithms to be employed.

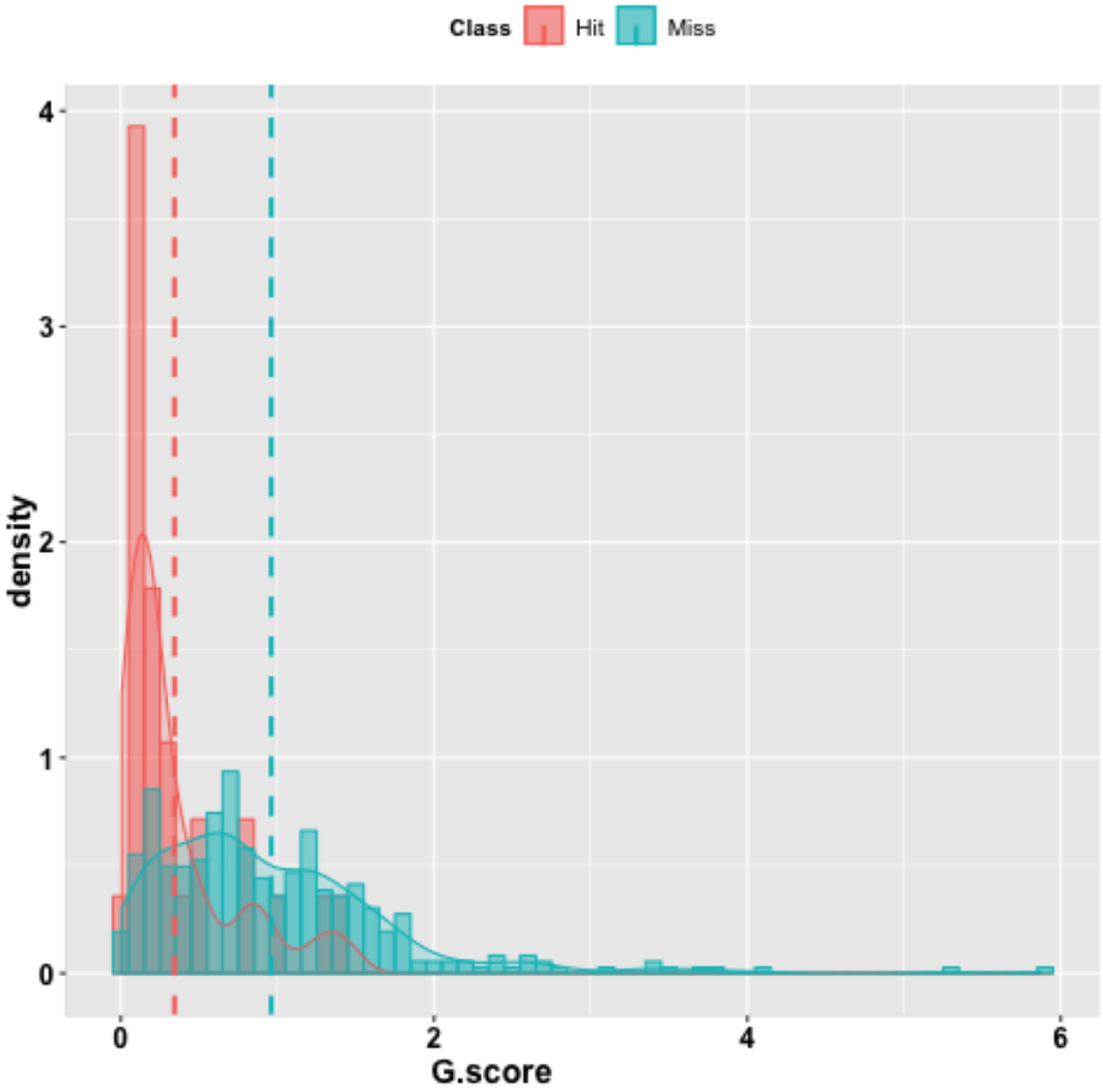


Figure 4·1: Distribution of G scores from sequence #6 classified as hits and misses. The histogram for each class of GAGfinder result shows that G scores are overall lower for hits than they are for misses.

One goal that we had for GAGrank was to be able to assign statistical significance to each possible structure, allowing for a significance cutoff to filter out unlikely structures. GAGrank does not currently deliver this kind of numerical significance value, but there are ways it could be done. For one, we could shuffle the edges of the network, keeping the same distribution of node degree and edge weight, and then re-analyze the shuffled networks. The scores for each structure in each shuffled network themselves would form a distribution against which the score for each structure in the real network could be compared. Assuming the distribution of shuffled network scores follows a typical statistical distribution (such as a normal distribution), we can estimate the probability that the real structure's score is from that distribution, or its p-value. A caveat to any GAG sequencing method that enumerates every possible structure and compares its possible fragments to those found in a MS^2 spectrum is that there is substantial overlap between closely related structures. This means there would likely be several structures whose p-values would surpass the *a priori* significance threshold. In these cases, the overall ranking would represent the best quantity for distinguishing structures.

Generalizing these methods to handle peptide or branched glycan MS^2 data would be difficult. As previously mentioned, the number of possible peptides grows exponentially with sequence length, making a brute force method for peak finding like GAGfinder untenable. For branched glycans, the nonstandard branching patterns that occur would increase search space for fragments, slowing down the process and making it more convoluted. Given a list of found fragments for peptides or branched glycans, it would be interesting to see how a GAGrank-like method would perform. However, methods for sequencing these biomolecular classes are already well-established, so there is no need for new techniques.

4.3 GAG Sequencing Using Enrichment Analysis

In my journey toward developing GAGrank, I first experimented with using a gene set enrichment analysis (GSEA)-like approach. GSEA [130] is a systems biology method whereby sets of genes are tested for differential expression, rather than individual genes. In biology, nothing happens in a vacuum. Indeed, the machinery involved in living systems is considerably interrelated, and often, sets of genes change in tandem to cause or in response to changes in phenotype. With the advent of high-throughput nucleic acid sequencing methods, it is now possible to test for changes in expression across an entire genome at once. GSEA tests for differential expression in gene sets by ranking the genes in order of a particular test statistic (such as a t-value) and assessing whether the set is clustered at the top or bottom of the list. Here, a set could mean any group of genes that works in tandem, from a known genetic pathway to a group of genes associated with a particular gene ontology (GO) term.

GSEA uses a random walk along the ranked list of genes to determine whether the genes in a set are distributed randomly along the list, or if they cluster at the top or bottom. As the walk progresses, a running score – the enrichment score (ES) – is tabulated, and it increases when it reaches a gene in the current set, and it decreases when it reaches a gene not in the current set. GSEA calculates the ES starting at both the top and bottom of the ranked list, so as to check for enrichment in upregulated and downregulated gene sets. The final ES is the maximum deviation from zero in the random walk, and it approximates a weighted Kolmogorov-Smirnov statistic. In order to assess significance of the ES, GSEA permutes the gene names a number of times and produces a distribution of possible ESs against which the real ES can be calculated. Finally, multiple testing correction procedures are performed to decrease the chance of false positive matches.

There are obvious analogs to the parts of GSEA in GAG MS^2 analysis. The

genes ranked by test statistic are analogous to the output of GAGfinder, sorted by G score from lowest to highest (recall: a lower G score represents a better fit between theoretical isotopic distribution and experimental isotopic distribution). The gene sets are analogous to the possible fragments for a particular sequence. The genome is analogous to the fragments found in the spectrum. With this in mind, I encoded the GSEA algorithm for GAG MS^2 data for GAG structural determination and tested its efficacy on a set of ten synthetic GAG saccharides.

Table 4.1 shows the results of the GSEA-like approach on the saccharides shown in Figure 3·3A. The main takeaway is the approach's uneven performance on these standards. We would expect a similar performance for the GSEA-like approach as that of GAGrank, but the rankings appear to be random. Therefore, we can conclude that for GAGs, testing for enrichment performs no better than randomly selecting a structure for a given composition.

Table 4.1: GSEA-inspired method performance for the GAGrank training compounds. See Figure 3·3A

Test Compound	Ranking	% incorrect outscored
#1	#2 out of 2	0%
#2	#1,071-#1,097 out of 1,848	40.7%
#3	#88-#114 out of 440	74.3%
#4	#1,361-#1,372 out of 1,584	13.4%
#5	#3 out of 60	96.6%
#6	#80-#106 out of 1,848	94.3%
#7	#38-#46 out of 990	95.4%
#8	#870-#872 out of 3,640	76.1%
#9	#8,091-#8,258 out of 23,298	64.6%
#10	#142-#144 out of 1,092	86.9%

There are a couple of main culprits for why this approach fails for GAG sequencing. One problem is the substantial overlap between fragment sets for closely related sequences. This prevents the approach from being able to significantly separate structures along the ES axis. Another problem is the nonstandard size of fragment sets for sequences. Consider sequence #8 in Figure 3·3A. The disaccharide composition

with one GlcA and one GlcNS occurs seven times in the structure. The set of possible fragments for this structure does not include that composition seven times, so as to avoid redundancy, so this reduces the structure's fragment set size. Since fragment set size is a parameter in the algorithm and having a smaller set size improves the ES, we can say that the method is biased toward structures with a lot of redundancy. Indeed, that structure had one of the better performances during testing, which can be attributed to its smaller complement of possible fragments.

4.4 Future Work

There are four obvious next steps for these projects. The first is integrating GAGfinder and GAGrank into a software pipeline and graphical user interface that reduces user involvement and therefore user error. The second is utilizing machine learning to assist in uronic acid C5 epimer determination. The third is statistical analysis of the GAGrank results using the earlier-described permutation test. The fourth is developing a pipeline for processing LC-MS^2 data.

As they currently exist, GAGfinder and GAGrank are exclusively command line interfaces with many required and optional arguments. This presents a learning curve for researchers who are not familiar with or comfortable in a command line setting. A natural way to fix this problem is to integrate the software into a complete pipeline that has an associated graphical user interface. This allows researchers to operate in a more conventional window setting. Furthermore, the researcher would only need to complete one step, as the output from GAGfinder would be streamlined into GAGrank. This would reduce user error. Since both programs are written in Python, the use of modules such as flask would make this possible, and in web format.

The problem of uronic acid C5 epimer determination in GAG sequencing is fundamentally a classification problem, and can be handled using machine learning clas-

sification methods. For instance, for a HS octasaccharide, there are four uronic acid residues, each of which is either GlcA or IdoA. Based on the spectral data, it may be possible to classify each of these as GlcA or IdoA. One such classification method is a decision tree, where the answers to a series of questions lead the classification procedure through a branched tree structure. Furthermore, the random forest procedure generates a number of decision trees at runtime and selects the one that performs the best. This is to combat decision trees' tendency to overfit data. A decision tree for uronic acid determination would most likely test for the presence or absence of particular fragments to make a prediction of each uronic acid. While these two methods are examples of classification procedures that could be used for this problem, any number of them could also be effective.

While GAGrank demonstrates excellent performance on GAG synthetic saccharides, it does not provide statistical significance in the form of p-values. As described earlier, one way to achieve this would be a permutation test of the edges in the bipartite network. This would entail keeping all the existing edges and their edge weights but changing the node at one or both ends of the edge. In order to avoid accidentally biasing the test, we would need to keep the degree distribution constant for each node. After each permutation, the BiRank algorithm would be re-run, and the scores for each sequence would be tabulated into a distribution of scores against which the score for the real network could be tested. This would allow for the assignment of a p-value for each possible sequence.

Finally, GAGfinder and GAGrank have shown success in their stated goals in a proof-of-principle fashion, using only tandem mass spectra of pure, synthetic GAG standards. When dealing with actual biological data, the data will come from an LC-MS^2 experiment, which results in thousands or millions of mass spectra. This represents a substantial increase in computational complexity over the single spectrum

analyses shown in Chapters 2 and 3. In order to be able to handle the excess data from an LC-MS^2 experiment, more nuanced and clever approaches need to be developed. For one, parallelization using a computing cluster will reduce the amount of time spent analyzing spectra, since several could be analyzed at a time. Furthermore, Python – in which GAGfinder and GAGrank are developed – uses dynamic bindings, while its faster contemporaries such as C++ or Java use static bindings, meaning that each Python function call requires a string lookup. This adds significant computing time to the software, and a translation from Python language to a static binding language would speed up the process.

4.5 Conclusion

In conclusion, the methods described in this dissertation expand the knowledgebase of GAG sequencing and provide researchers two valuable tools to achieve this end. It is my hope that these methods will allow researchers to study GAG sequence more in depth and be able to tease out exactly how GAG sequence affects biological processes. For instance, since GAG sequence is so strongly associated with cell-cell interactions, which are key for cancer proliferation, the fine structure could potentially be a biomarker for different types of cancer.

Appendix A

PEAKSviz: Data visualization and statistical analysis of PEAKS proteomics data

A.1 Introduction

As its name implies, label-free quantification (LFQ) is a proteomics method for quantifying proteins in a set of biological samples without the use of an isotopic labeling tag. In a liquid chromatography/tandem mass spectrometry (LC-MS^2) experiment, the MS^2 stage is for peptide identification, while the MS^1 stage is for peptide quantification. LFQ extracts precursor ion signal at the MS^1 level for abundant peptides. These peptides, and their associated abundances, can then be mapped back to protein sequences using sequence alignment techniques.

PEAKS (Bioinformatics Solutions, Waterloo, ON, Canada) is a software suite that assists researchers in performing all of these tasks. In a PEAKS proteomics experiment, researchers upload tandem mass spectrometry data for their samples, and they can download a comma-separated variable (CSV) file containing this information. There are a number of input parameters, such as identification false discovery rate

(FDR) and the number of unique identified peptides required for protein identification, among others. These parameters are determined *a priori* by the researchers, and depend on their preferences for how narrow they want the search to be.

We developed a web application, PEAKSviz, in R Shiny for data visualization and statistical analysis of PEAKS proteomics output data. PEAKSviz is freely available to use and its source code is publicly available. PEAKSviz produces heatmaps, principal component analysis (PCA) plots, hierarchical clustering dendrograms, and differential analysis at either the protein or the peptide level.

A.2 PEAKSviz overview

There are two possible analysis types for PEAKSviz: protein-level and peptide-level. Each of these analysis types produces the same data visualizations and statistical analyses on different PEAKS outputs. The protein-level analysis uses the PEAKS output file proteins.csv as input, while the peptide-level analysis uses the PEAKS output file protein-peptides.csv as input. The steps of the pipeline are described below, and screenshots from PEAKSviz are shown in Figure A·1. Data used to generate the visualizations in Figure A·1B-D are described in [131], and can be downloaded for testing purposes via the PRIDE [101] public repository with data set identifier PXD008990.

Step 1: Load data – In this step, the user uploads the appropriate PEAKS output file as input into the pipeline and inputs several other pieces of information about the project: the project name, imputation method, number of different groups in the data, and a protein or peptide list for subsetting the data. Due to the significant missing data problem in LFQ experiments, PEAKSviz imputes the missing values by one of two methods: median imputation, where each missing value is converted to the median intensity for the corresponding sample, or k-nearest neighbors (KNN)

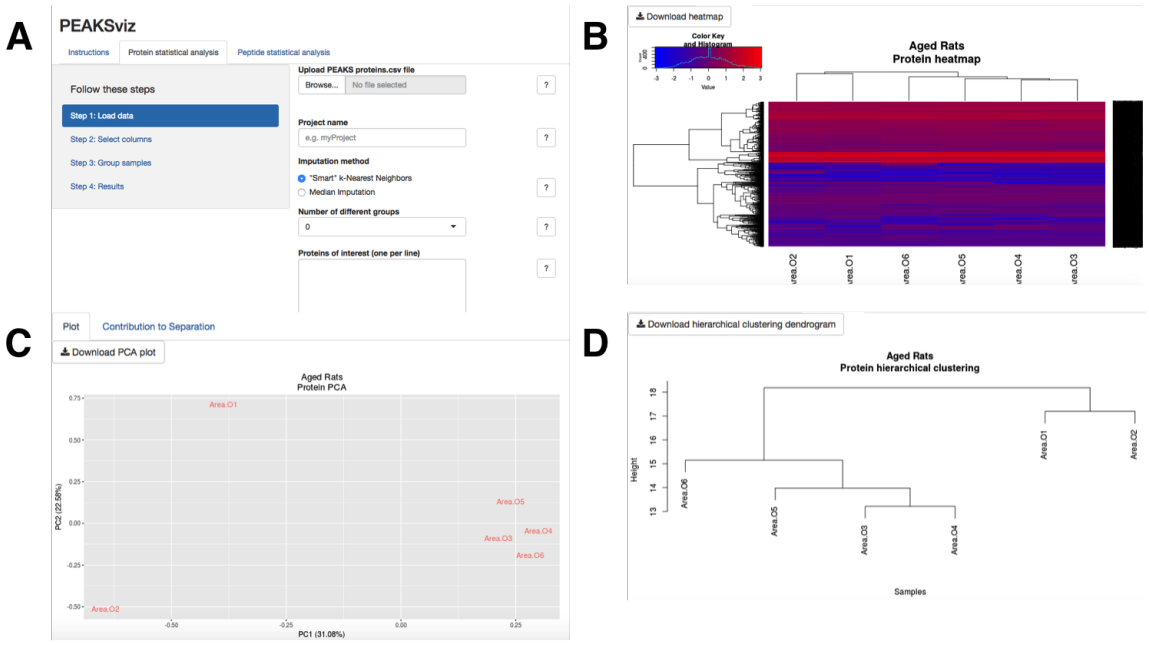


Figure A·1: Screenshots from the PEAKSviz user interface. A. Data input page, including inputs for the PEAKS CSV file, project name, imputation method, group number, and proteins for which to filter data. B. Heatmap on logged, scaled protein expression data. C. PCA plot for logged, scaled protein expression data. D. Hierarchical clustering dendrogram for logged, scaled protein expression data.

imputation, where each missing value is converted to the average intensity for that protein across the k most similar samples. The number of different groups corresponds to the number of different types of sample in the project (e.g. two groups, if comparing disease to healthy). Finally, the data can be subsetted to a protein or list of proteins. These could be proteins associated with a gene ontology (GO) term, or a biochemical process. For the peptides analysis, only one protein may be selected to subset the data.

Step 2: Select columns – In this step, the user tells PEAKSviz what columns in the file correspond to the intensity data for the proteins or peptides. PEAKSviz attempts to guess at which columns the user wants by searching the column names for columns containing the word "area." This is due to the tendency of test users to label the samples' intensities as areas, for the area under the peak.

Step 3: Group samples – In this step, the user tells PEAKSviz which samples belong in which group, if the user designated in step 1 that there are groups in the data. This is for appropriate color-coding for the PCA diagram in step 4.

Step 4: Results – Here, the results of the analyses are produced. There are four or five outputs, depending on if the user designated in step 1 that there are groups in the data or not. These outputs are the scaled and imputed intensity data, a heatmap, a PCA plot and contributions to principal component (PC) variation, a hierarchical clustering dendrogram, and differential expression.

A.3 Implementation

PEAKSviz was developed in the R statistical language [132] and the corresponding user interface was developed using the R package Shiny [133]. The KNN imputation procedure utilizes the impute.knn function of the Bioconductor package impute [134]. The heatmaps are rendered using the heatmap.2 function of the R package

gplots [135]. The PCA plots are rendered using various functions from the R package ggfortify [136, 137]. Every other step in the algorithm is performed using a base R function.

A.4 Conclusion

Here we have presented PEAKSviz, a web application for data visualization and statistical analysis of protein expression data. PEAKSviz works on any modern web browser, and requires only basic user interaction. PEAKSviz is perfect for researchers who wish to harness R's statistical data visualization techniques without needing to learn the language. We hope that other groups use PEAKSviz for their proteomics statistical analysis needs.

Appendix B

Supplemental Material for Chapter 2

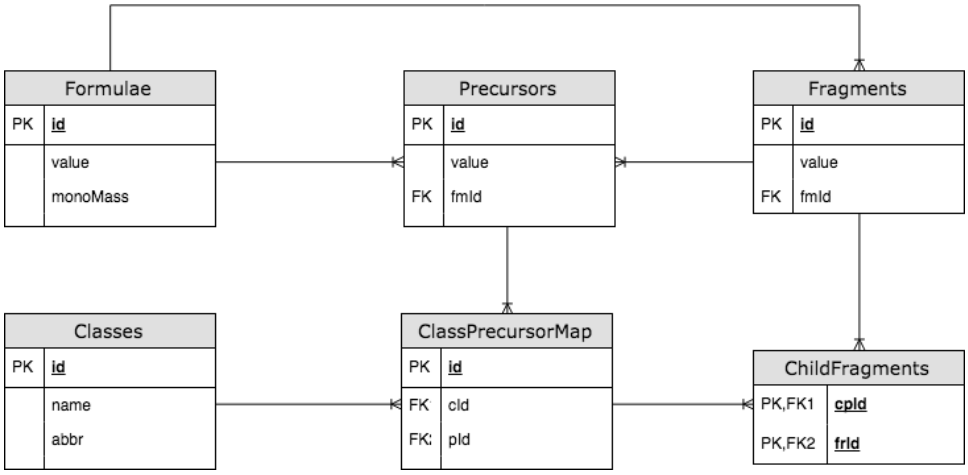


Figure B·1: Relational schema for GAGfragDB.

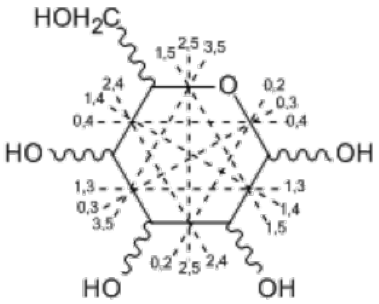
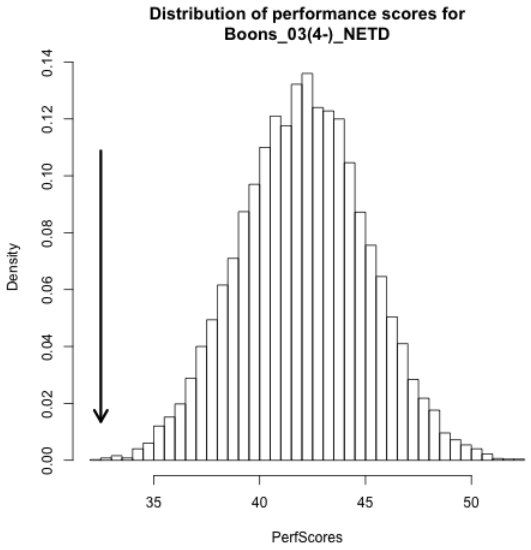
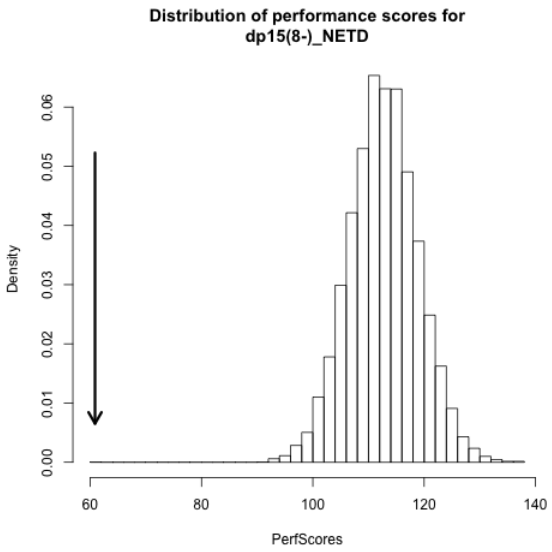


Figure B·2: Cross-ring cleavage patterns considered in GAGfinder.

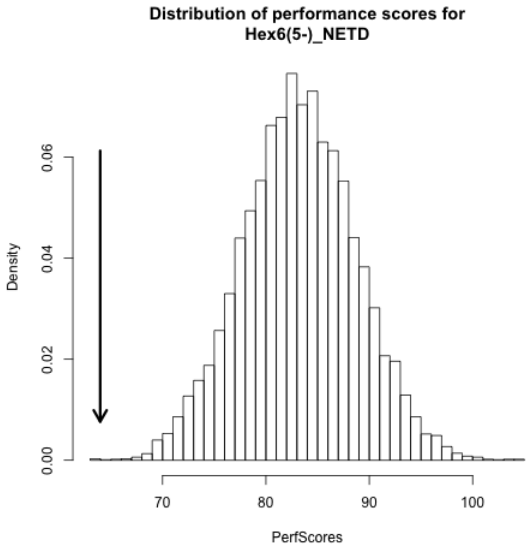
Figure B·3: Distributions of each saccharide's *PerfScore* permutation test. Arrows point to the *PerfScore* for each sequence.



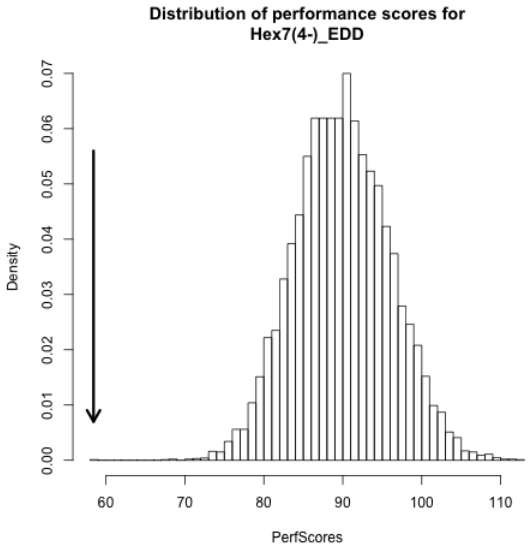
(a) Distributions of the permutation scores for saccharide #1 against the background.



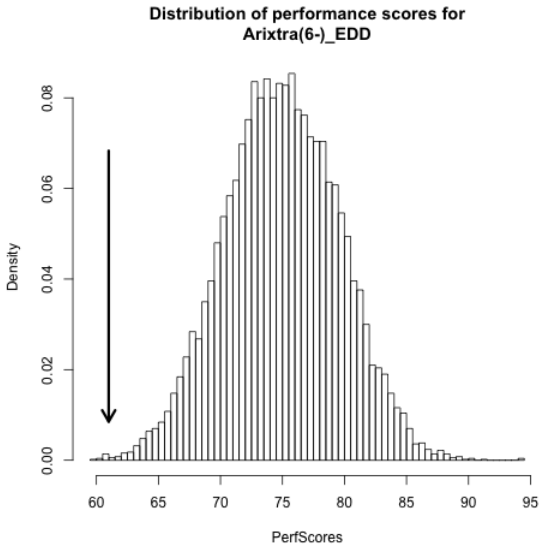
(b) Distributions of the permutation scores for saccharide #2 against the background.



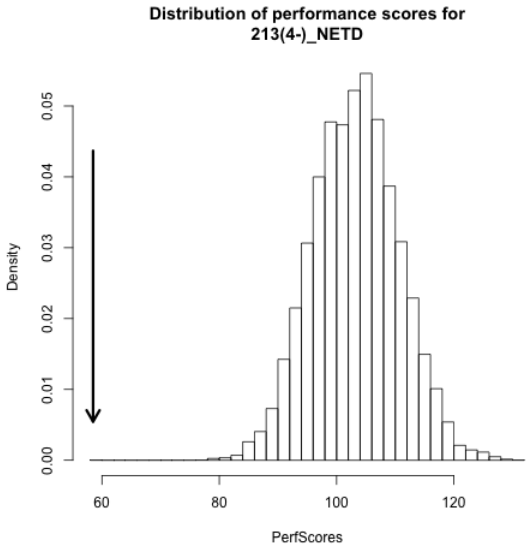
(c) Distributions of the permutation scores for saccharide #3 against the background.



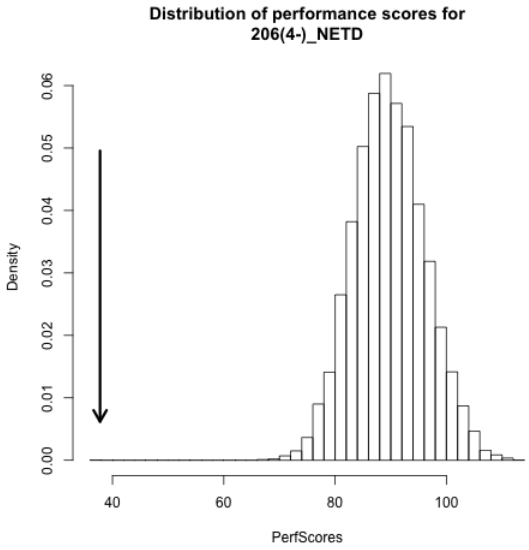
(d) Distributions of the permutation scores for saccharide #4 against the background.



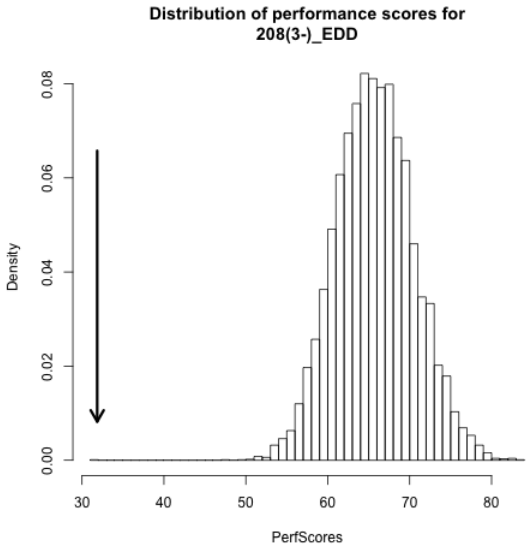
(e) Distributions of the permutation scores for saccharide #5 against the background.



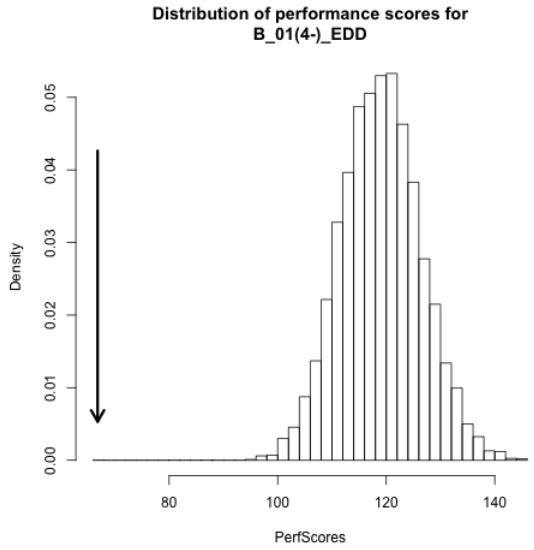
(f) Distributions of the permutation scores for saccharide #6 against the background.



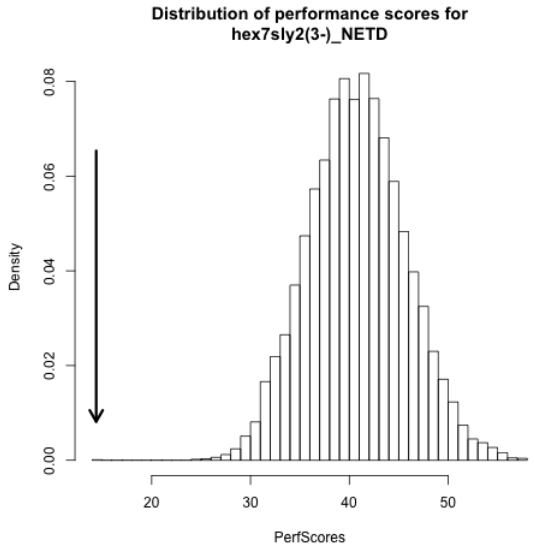
(g) Distributions of the permutation scores for saccharide #7 against the background.



(h) Distributions of the permutation scores for saccharide #8 against the background.

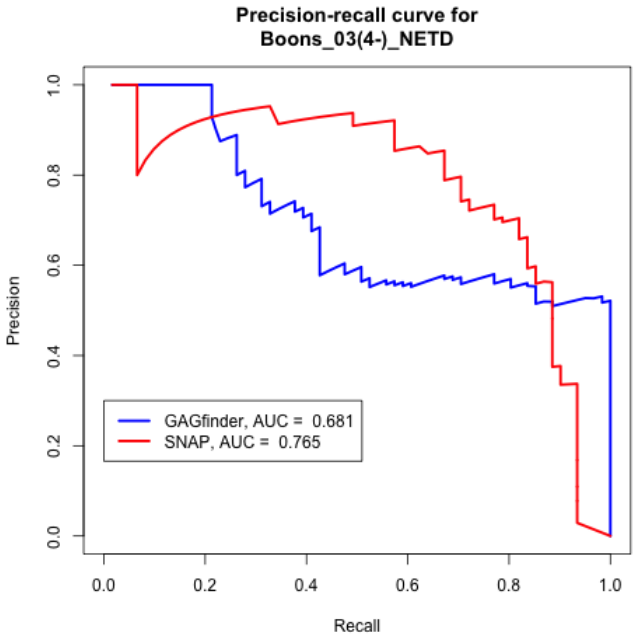


(i) Distributions of the permutation scores for saccharide #9 against the background.

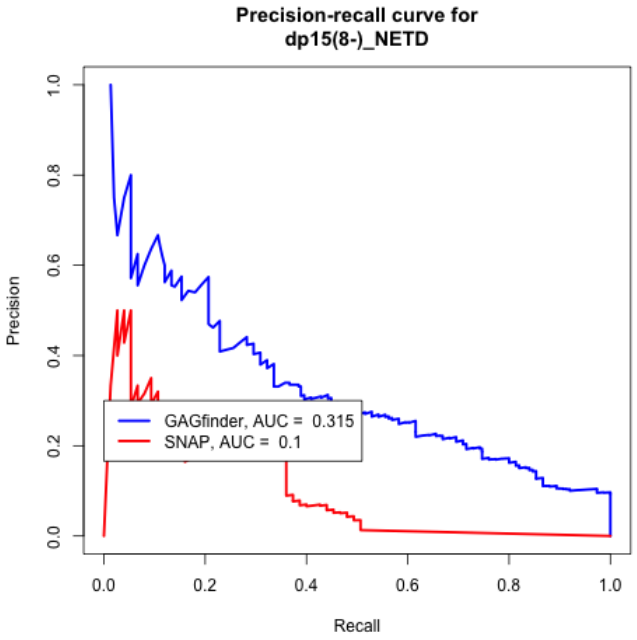


(j) Distributions of the permutation scores for saccharide #10 against the background.

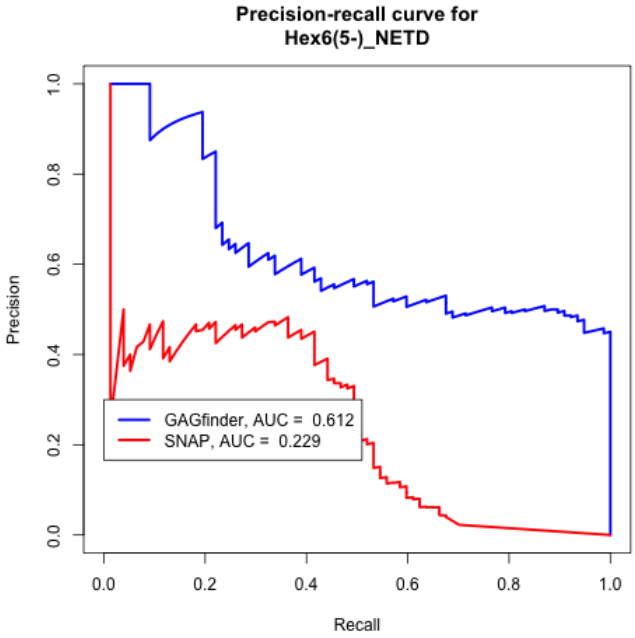
Figure B·4: Precision-recall curves for GAGfinder and SNAP for each saccharide.



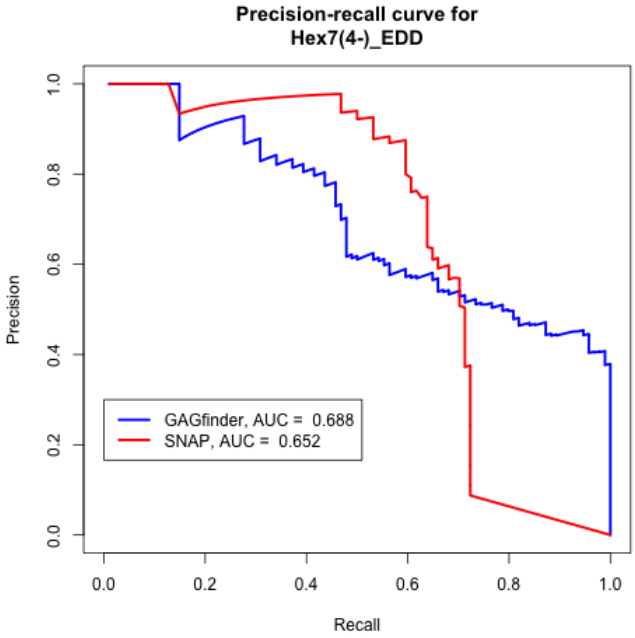
(a) Precision-recall curve for GAGfinder and SNAP for saccharide #1



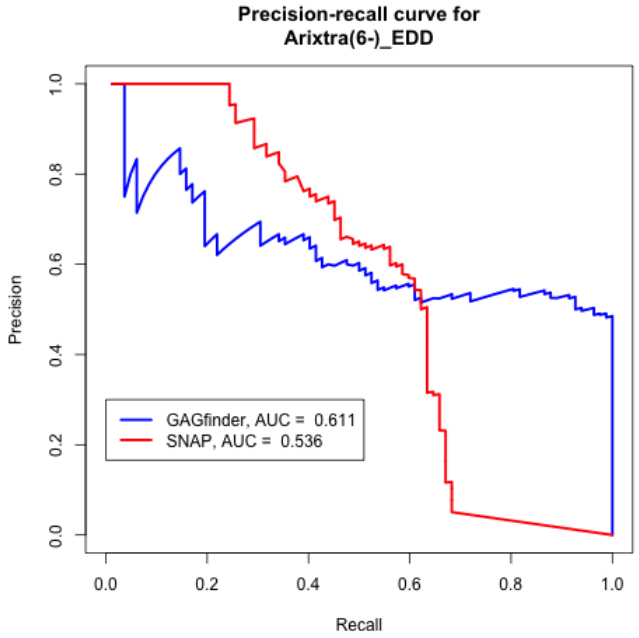
(b) Precision-recall curve for GAGfinder and SNAP for saccharide #2



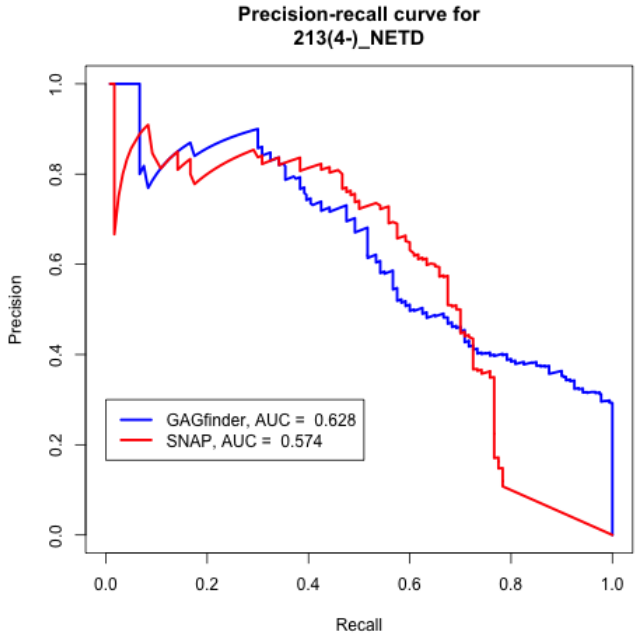
(c) Precision-recall curve for GAGfinder and SNAP for saccharide #3



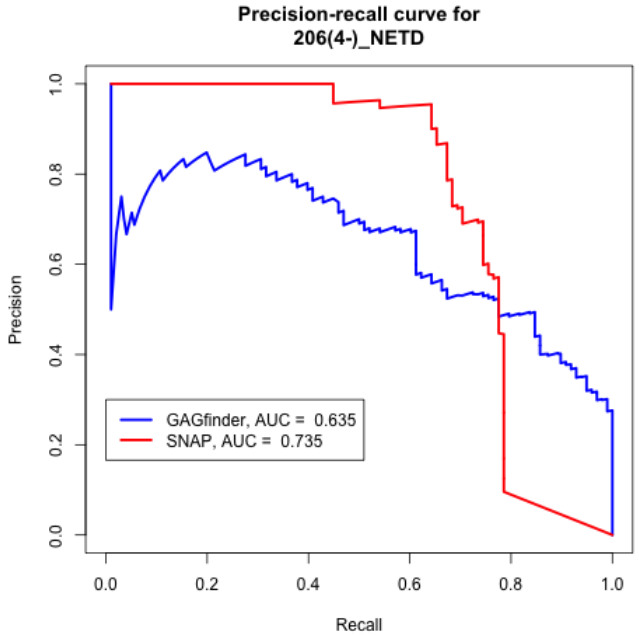
(d) Precision-recall curve for GAGfinder and SNAP for saccharide #4



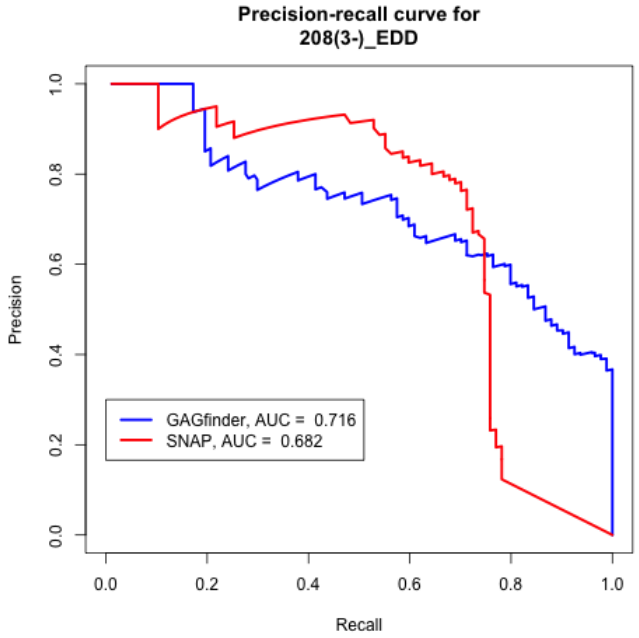
(e) Precision-recall curve for GAGfinder and SNAP for saccharide #5



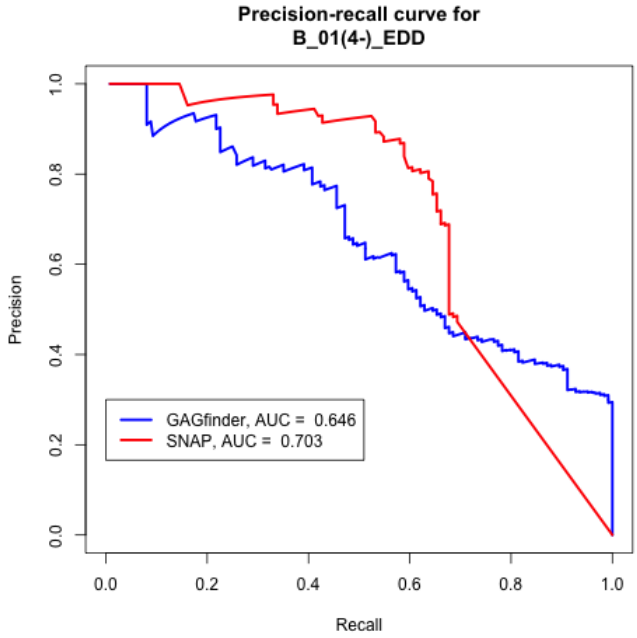
(f) Precision-recall curve for GAGfinder and SNAP for saccharide #6



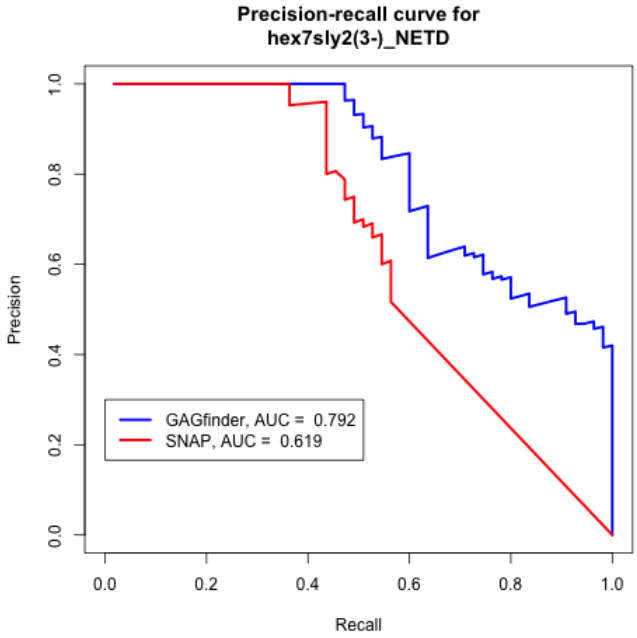
(g) Precision-recall curve for GAGfinder and SNAP for saccharide #7



(h) Precision-recall curve for GAGfinder and SNAP for saccharide #8

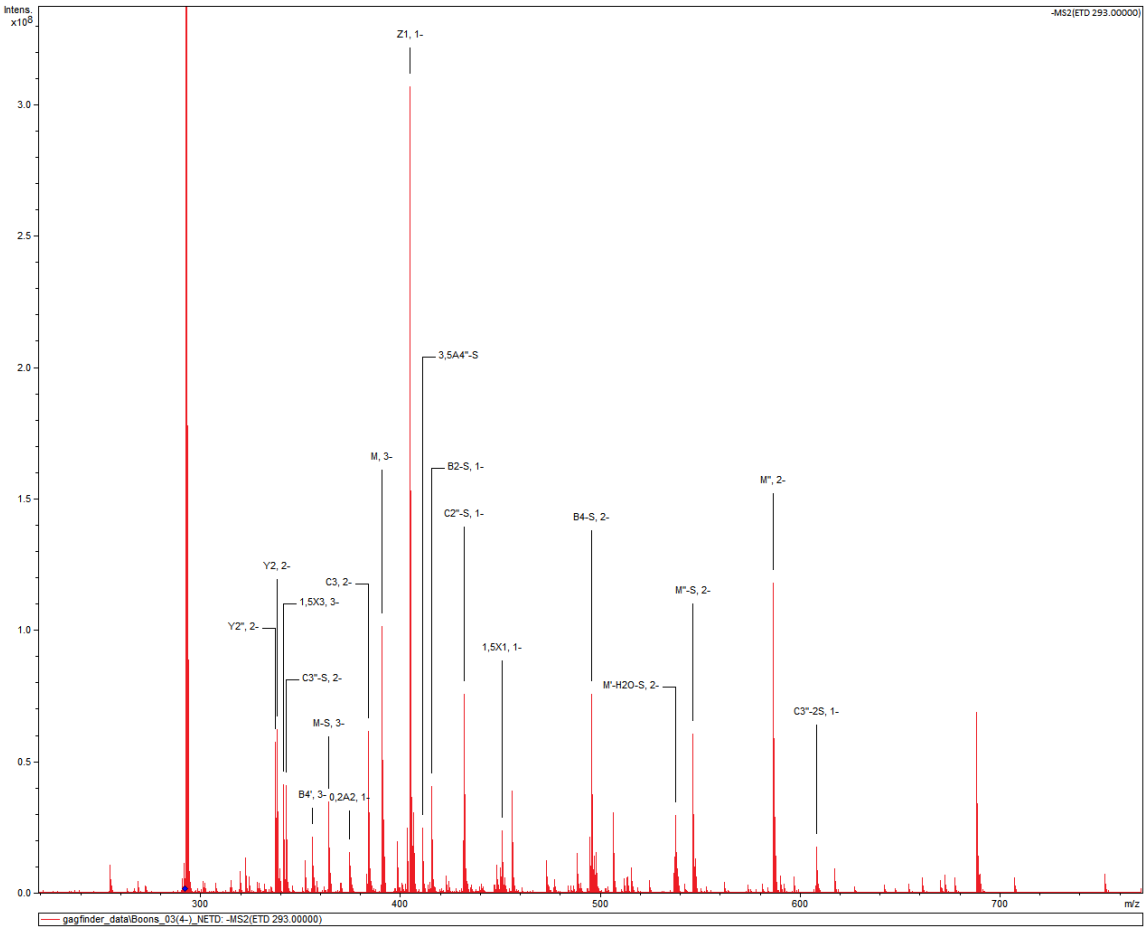


(i) Precision-recall curve for GAGfinder and SNAP for saccharide #9

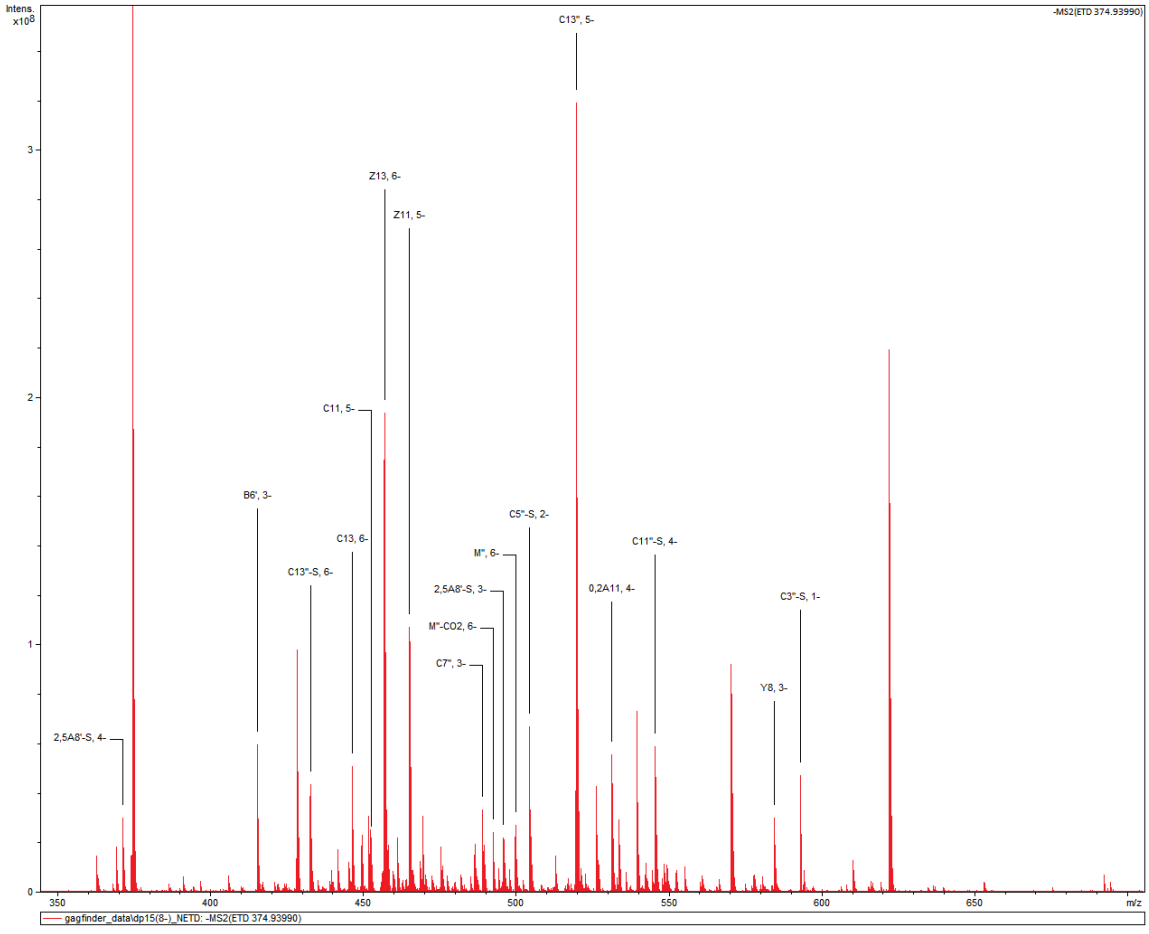


(j) Precision-recall curve for GAGfinder and SNAP for saccharide #10

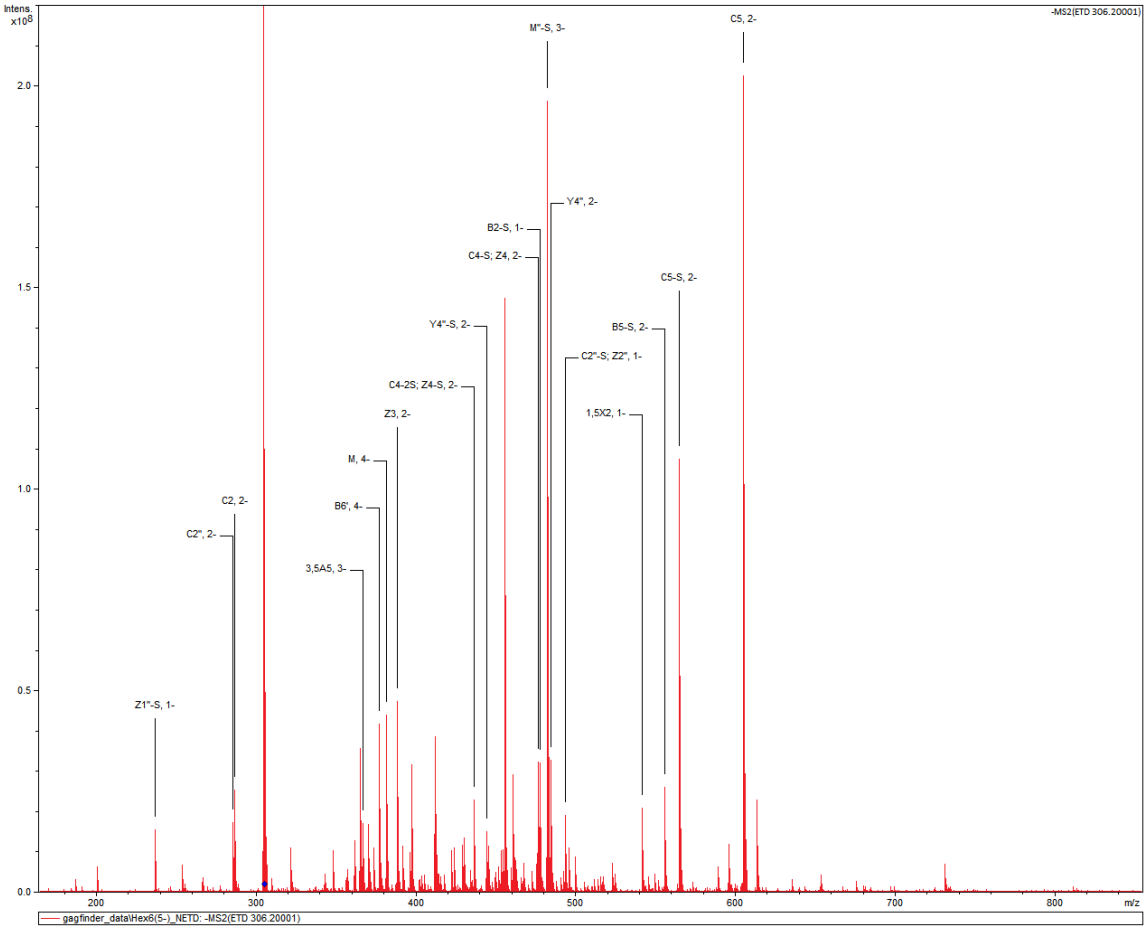
Figure B·5: Annotated spectra for each test saccharide.



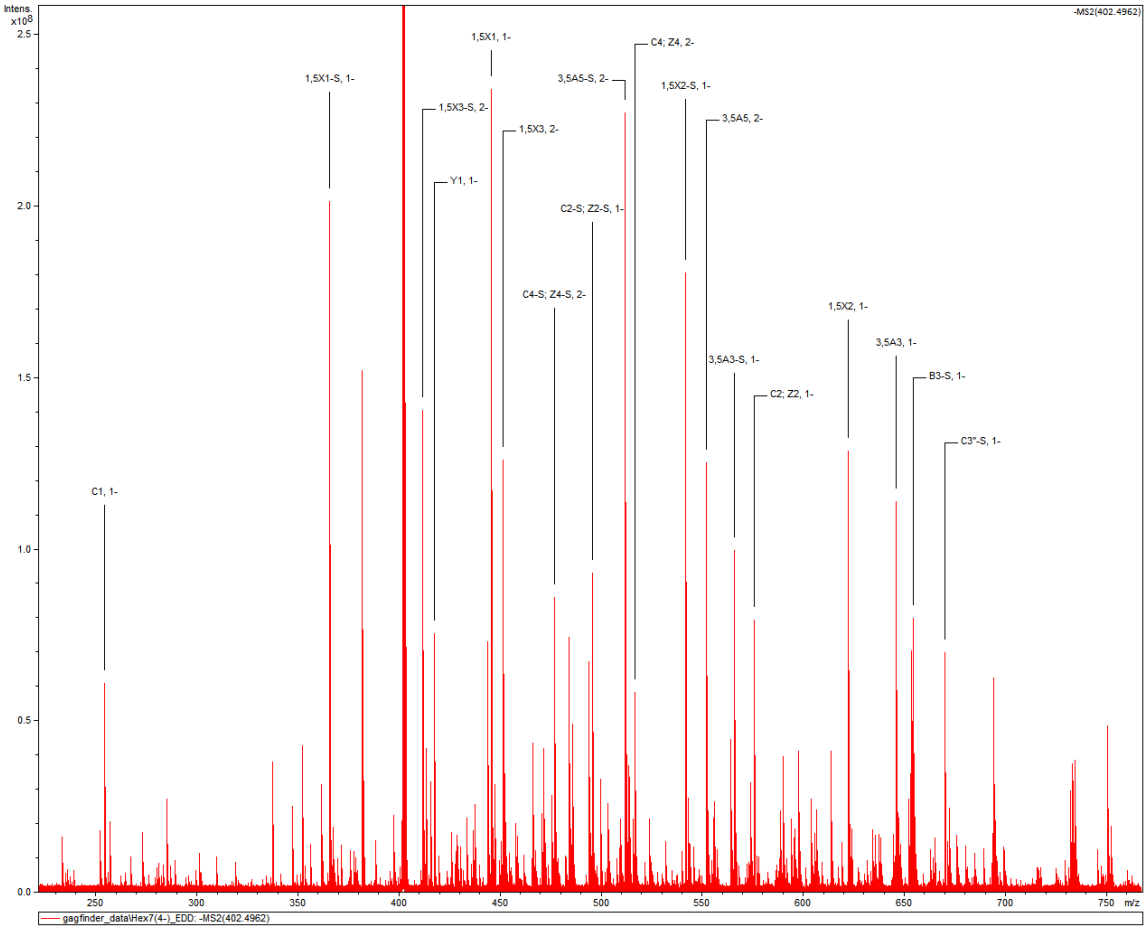
(a) Annotated spectrum for saccharide #1



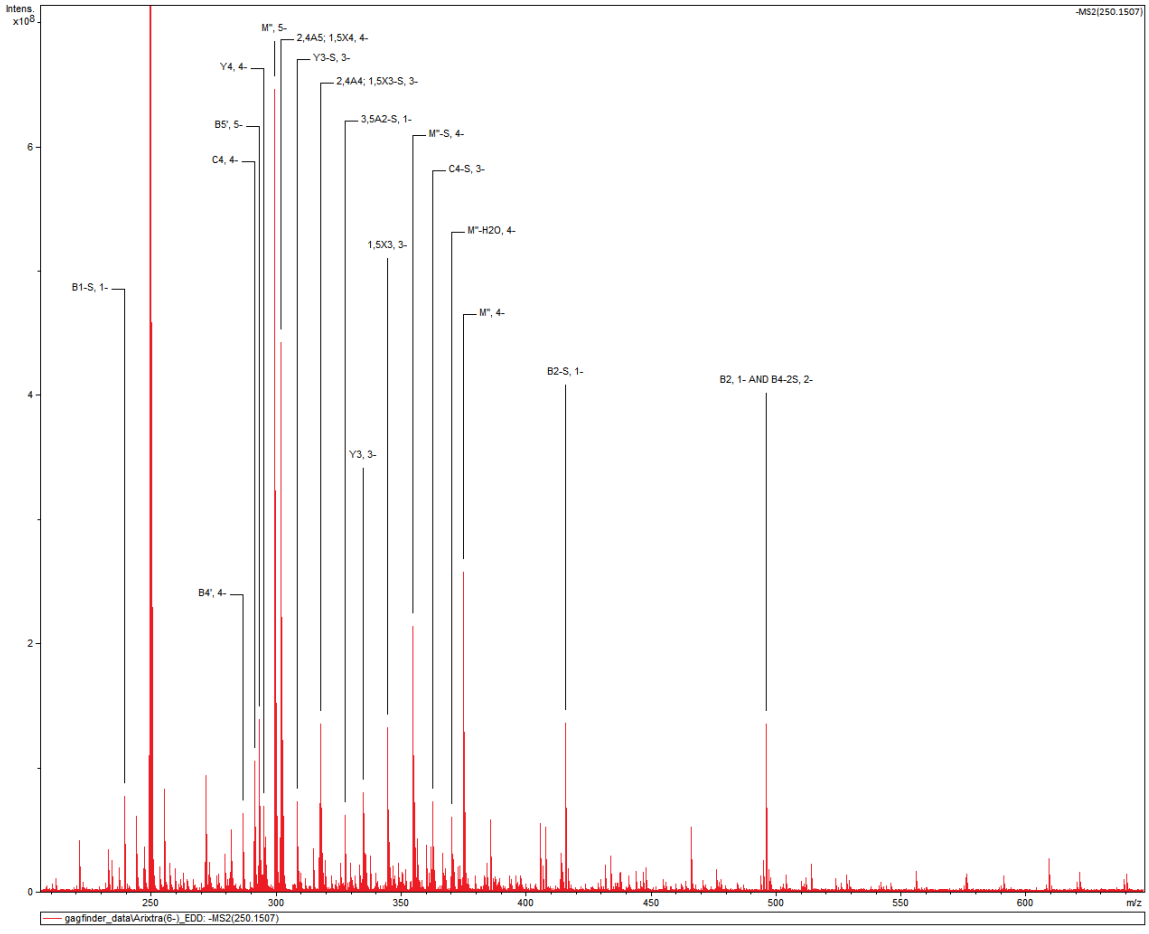
(b) Annotated spectrum for saccharide #2



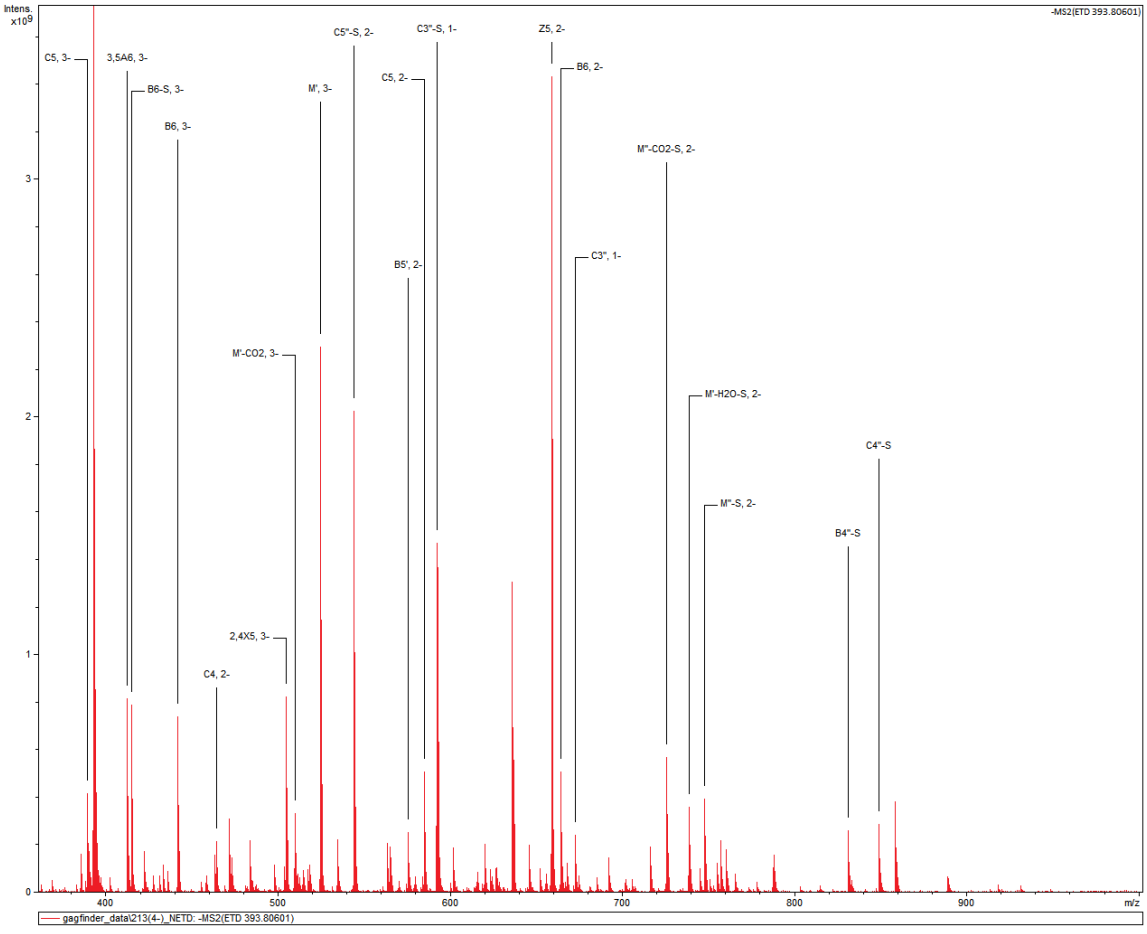
(c) Annotated spectrum for saccharide #3



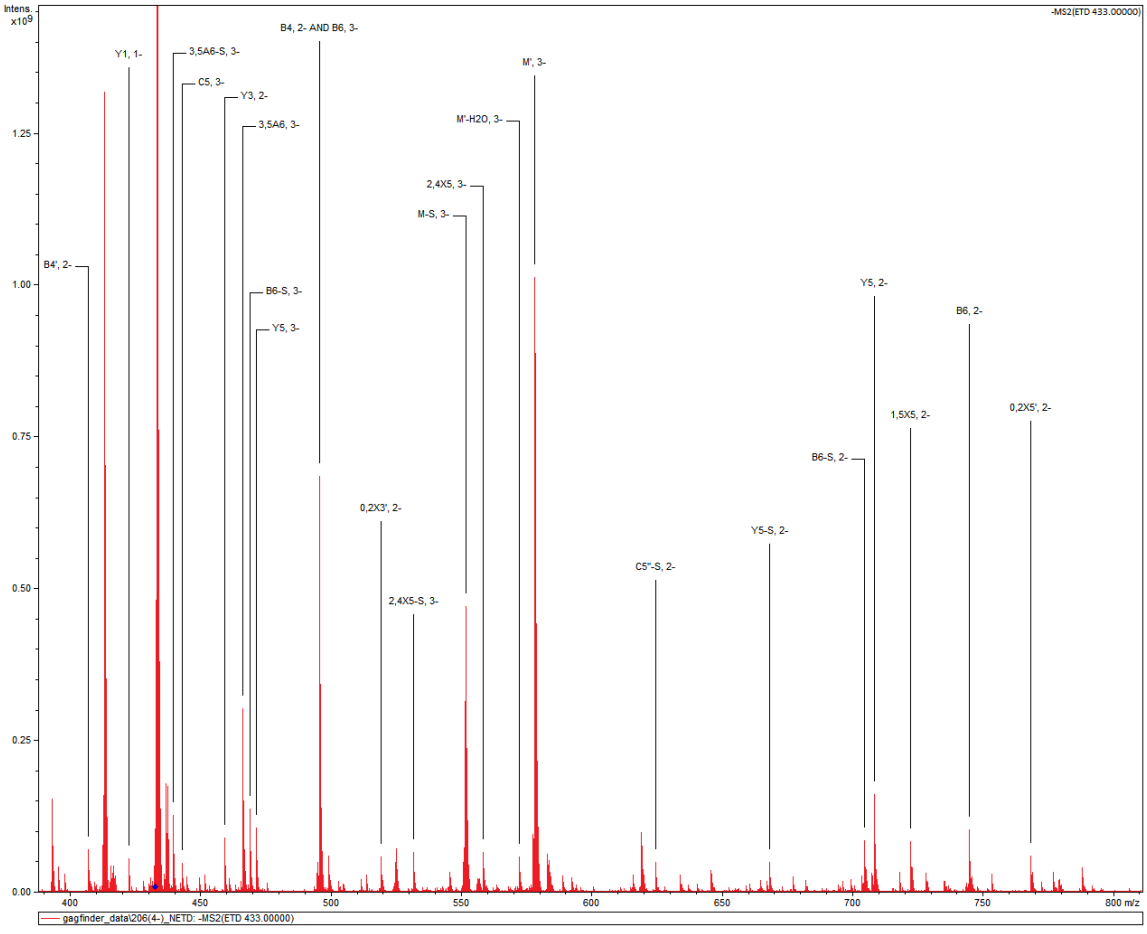
(d) Annotated spectrum for saccharide #4



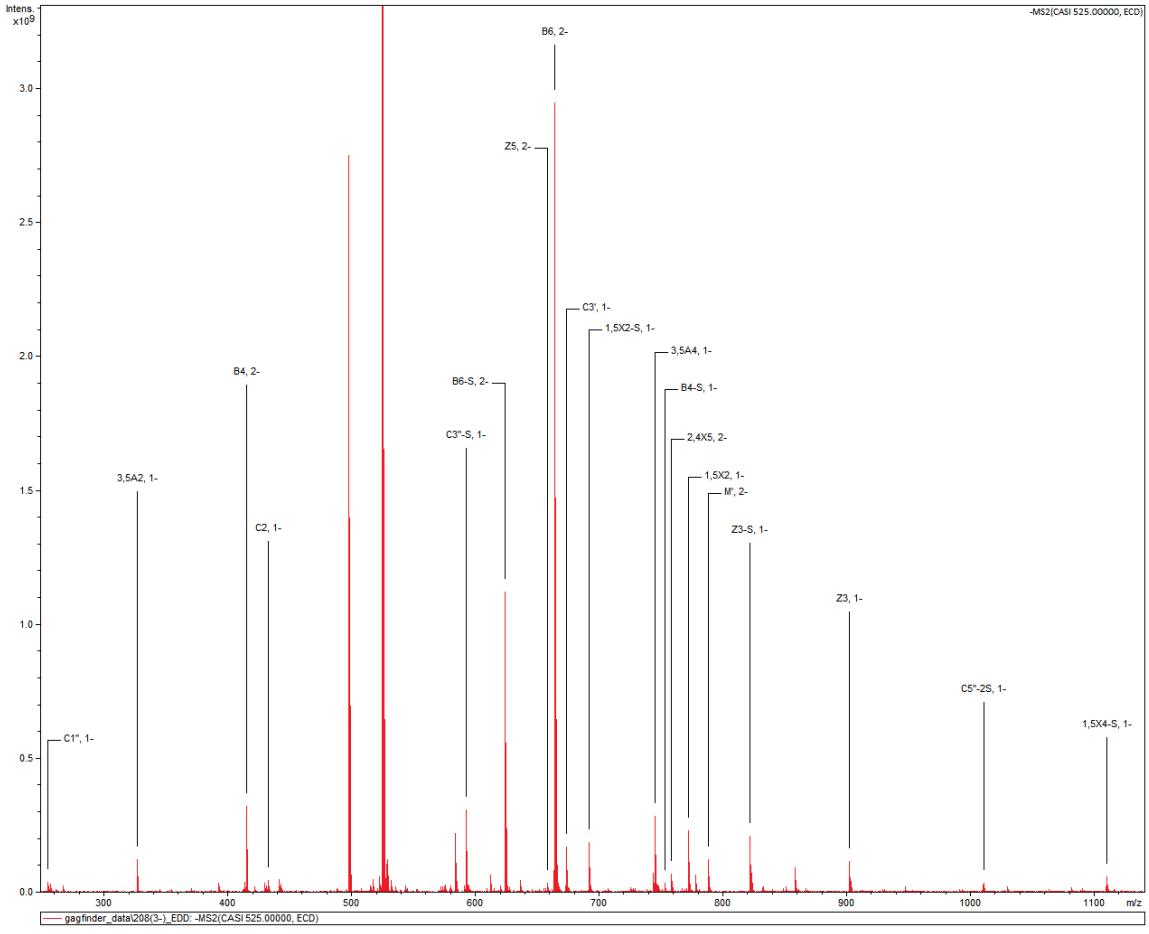
(e) Annotated spectrum for saccharide #5



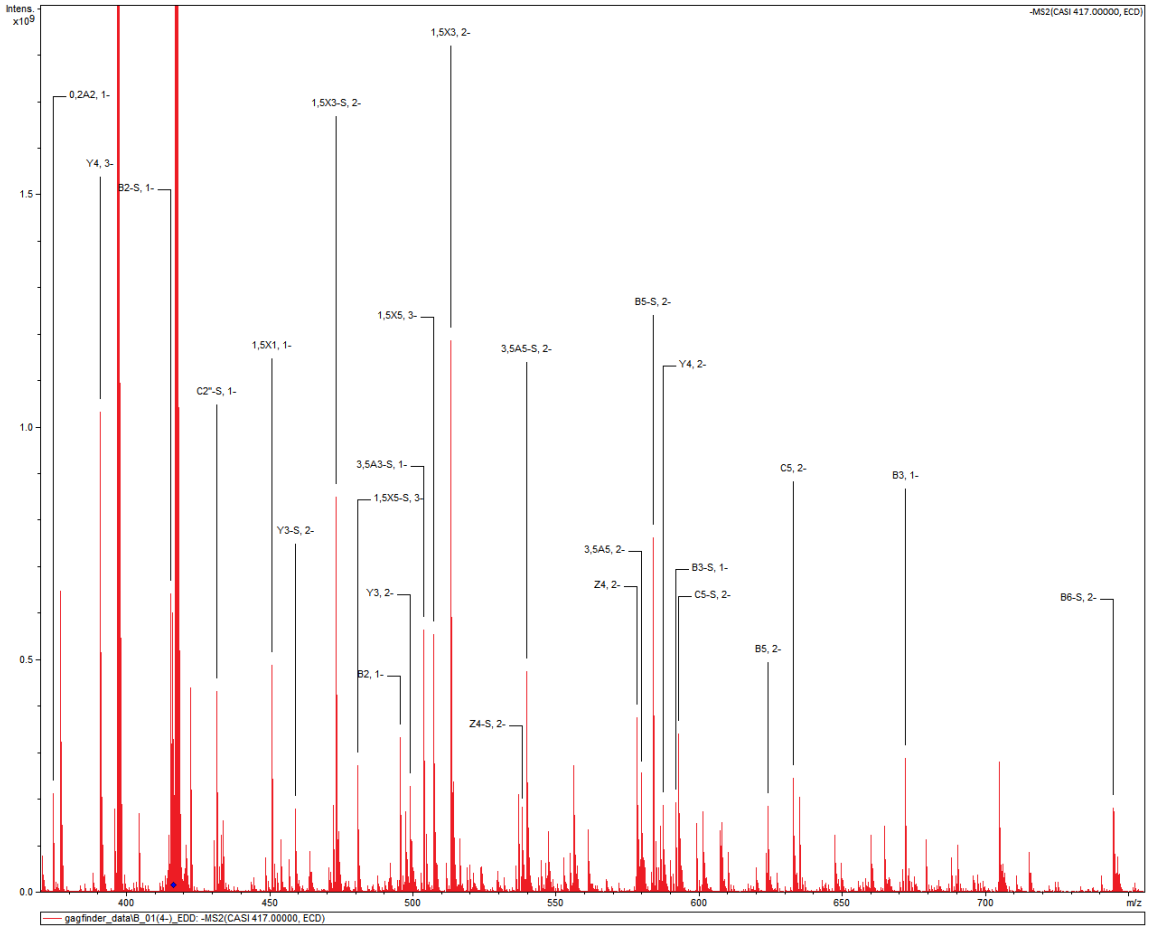
(f) Annotated spectrum for saccharide #6



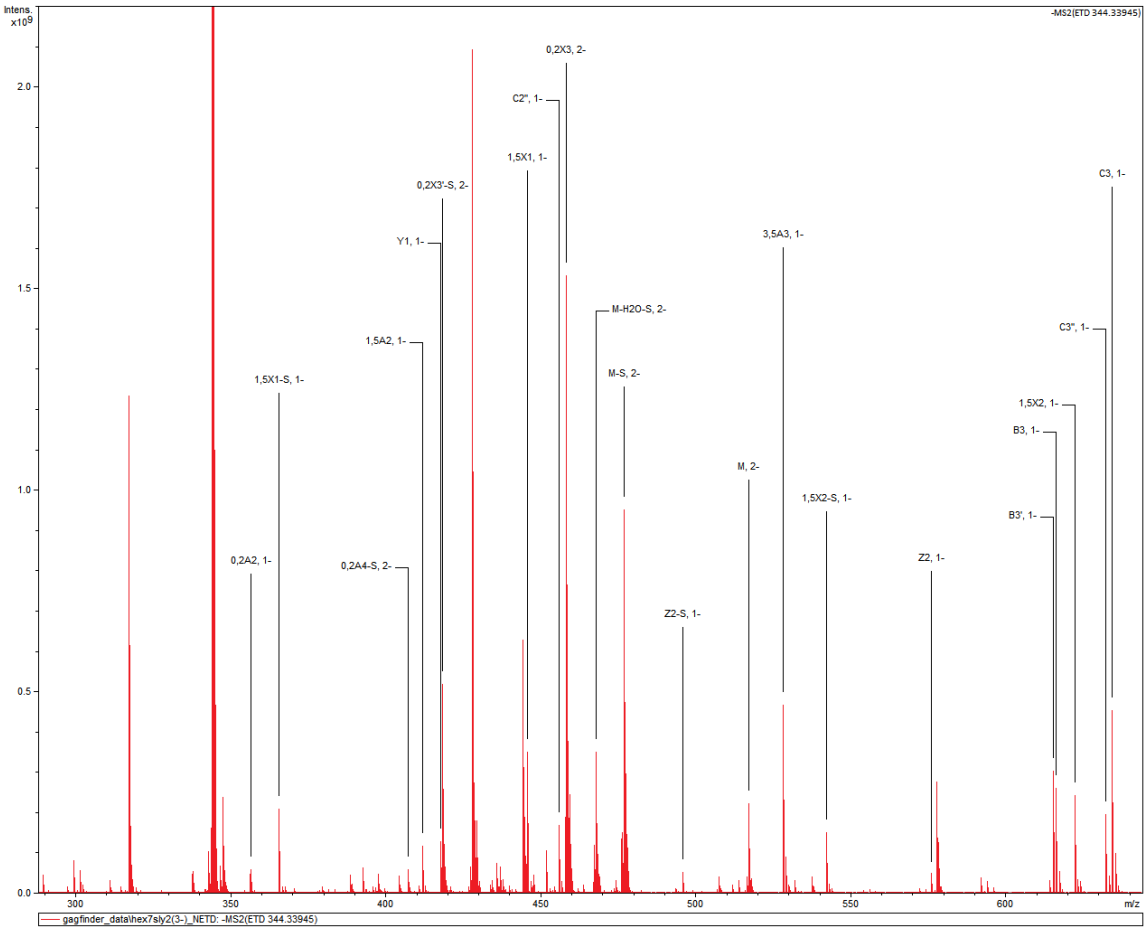
(g) Annotated spectrum for saccharide #7



(h) Annotated spectrum for saccharide #8

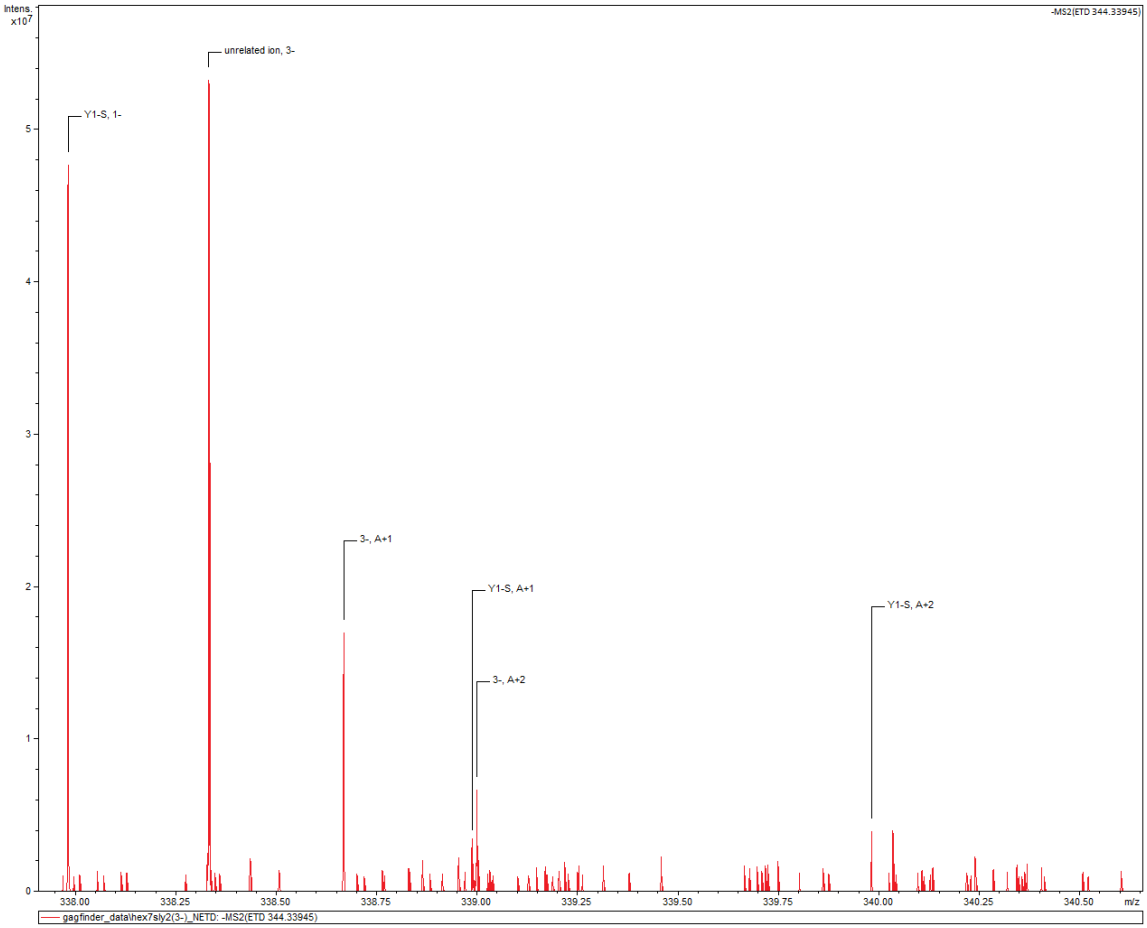


(i) Annotated spectrum for saccharide #9

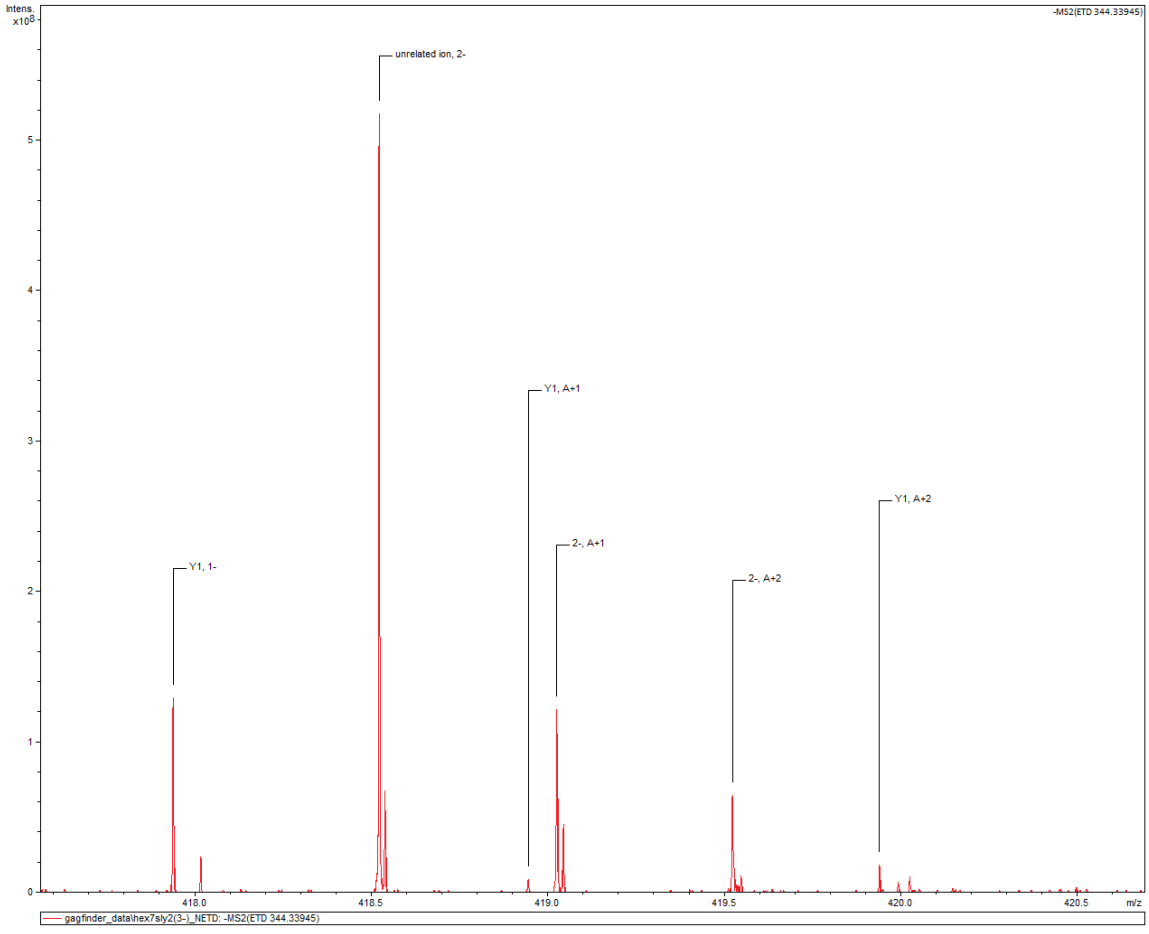


(j) Annotated spectrum for saccharide #10

Figure B·6: Zoomed in images of the Y1, 1- and Y1-S, 1- ions mentioned in Chapter 2. Notice how the ions are picked out of other high-intensity ions.



(a) Y1-S, 1-


Intens.
x10^8
-MS2(ETD 344.33945)
unrelated ion, 2-
Y1, A+1
Y1, A+2
2-, A+1
2-, A+2
Y1, 1-
5
4
3
2
1
0
418.0
418.5
419.0
419.5
420.0
420.5
m/z
gagfinder_data\hex7sly2(3-)_NETD: -MS2(ETD 344.33945)

(b) Y1, 1-

Appendix C

Supplemental Material for Chapter 3

Table C.1: Time data for each of the tested compounds. The two sections at the bottom represent the two isomeric mixtures tested.

Compound	Precursor z	Dissociation Method	Runtime (s)	# sequences	# fragments
Training #1	-2	NETD	4.711	2	19
Training #2	-4	NETD	9.614	1,848	50
Training #3	-3	NETD	4.207	440	29
Training #4	-5	NETD	8.647	1,584	57
Training #5	-4	NETD	3.426	60	37
Training #6	-5	EDD	12.913	1,848	93
Training #7	-4	EDD	8.654	990	54
Training #8	-5	NETD	16.128	3,640	38
Training #9	-6	NETD	171.189	23,298	15
Training #10	-4	NETD	16.127	1,092	48
Validation #1	-4	NETD	6.484	140	12
Validation #2	-5	NETD	8.454	1,584	54
Validation #3	-6	NETD	7.166	990	55
1A:1B, 100:0	-6	NETD	8.336	1,584	47
1A:1B, 90:10			9.588		51
1A:1B, 70:30			9.484		49
1A:1B, 50:50			9.629		45
1A:1B, 30:70			8.964		49
1A:1B, 10:90			10.303		43
1A:1B, 0:100			8.137		42
2A:2B, 100:0	-3	NETD	2.444	30	28
2A:2B, 90:10			1.512		27
2A:2B, 70:30			1.514		28
2A:2B, 50:50			1.588		32
2A:2B, 30:70			1.528		33
2A:2B, 10:90			1.518		28
2A:2B, 0:100			1.677		28

Table C.2: Training compound #1 results. Row with correct sequence is italicized.

Sequence	GAGrank score
HexNAc-HexA-HexNAc-HexA	*0.303482198*
HexA-HexNAc-HexA-HexNAc	0.242934209

Table C.3: Training compound #2 results. Row with correct sequence is italicized.

Sequence	GAGrank score
HexA-HexNS6S-HexA-HexNS6S-HexA-HexNS6S	*0.004031581*
HexA-HexNS6S-HexA-HexNS6S-HexA-HexNS3S	0.004028602
HexA-HexNS6S-HexA-HexNS3S-HexA-HexNS6S	0.004028602
HexA-HexNS3S-HexA-HexNS6S-HexA-HexNS6S	0.004028602
HexA-HexNS6S-HexA-HexNS3S-HexA-HexNS3S	0.004026018
HexA-HexNS3S-HexA-HexNS6S-HexA-HexNS3S	0.004026018
HexA-HexNS3S-HexA-HexNS3S-HexA-HexNS6S	0.004026018
HexA-HexNS6S-HexA-HexNS6S-HexA-HexN3S6S	0.004024705
HexA-HexNS6S-HexA-HexN3S6S-HexA-HexNS6S	0.004024705
HexA-HexN3S6S-HexA-HexNS6S-HexA-HexNS6S	0.004024705

Table C.4: Training compound #3 results. Row with correct sequence is italicized.

Sequence	GAGrank score
HexNS-HexA-HexNS-HexA-HexNS-HexA	*0.010001795*
HexNS-HexA-HexNS-HexA-HexN6S-HexA	0.009963715
HexNS-HexA-HexN6S-HexA-HexNS-HexA	0.009963715
HexN6S-HexA-HexNS-HexA-HexNS-HexA	0.009963715
HexNS-HexA-HexN6S-HexA-HexN6S-HexA	0.009937320
HexN6S-HexA-HexNS-HexA-HexN6S-HexA	0.009937320
HexN6S-HexA-HexN6S-HexA-HexNS-HexA	0.009937320
HexNS-HexA-HexNS-HexA-HexN3S-HexA	0.009927102
HexNS-HexA-HexN3S-HexA-HexNS-HexA	0.009927102
HexN3S-HexA-HexNS-HexA-HexNS-HexA	0.009927102

Table C.5: Training compound #4 results. Row with correct sequence is italicized.

Sequence	GAGrank score
HexA-HexNS6S-HexA2S-HexNS6S-HexA-HexNS6S	*0.003969474*
HexA-HexNS6S-HexA2S-HexNS3S-HexA-HexNS6S	0.003966378
HexA-HexNS3S-HexA2S-HexNS6S-HexA-HexNS6S	0.003966378
HexA-HexNS3S-HexA2S-HexNS3S-HexA-HexNS6S	0.003963694
HexA-HexNS6S-HexA2S-HexNS6S-HexA-HexN3S6S	0.003962329
HexA-HexN3S6S-HexA2S-HexNS6S-HexA-HexNS6S	0.003962329
HexA-HexNS6S-HexA2S-HexNS3S-HexA-HexN3S6S	0.003960183
HexA-HexNS3S-HexA2S-HexNS6S-HexA-HexN3S6S	0.003960183
HexA-HexN3S6S-HexA2S-HexNS3S-HexA-HexNS6S	0.003960183
HexA-HexNS3S-HexA2S-HexNS3S-HexA-HexN3S6S	0.003958323

Table C.6: Training compound #5 results. Row with correct sequence is italicized.

Sequence	GAGrank score
HexNS-HexA-HexNAc-HexA-HexNAc-HexA	*0.037645074*
HexN3S-HexA-HexNAc-HexA-HexNAc-HexA	0.036493308
HexN6S-HexA-HexNAc-HexA-HexNAc-HexA	0.036115857
HexN-HexA2S-HexNAc-HexA-HexNAc-HexA	0.032541920
HexNAc-HexA-HexNS-HexA-HexNAc-HexA	0.029770158
HexNAc-HexA-HexN6S-HexA-HexNAc-HexA	0.029618825
HexNAc-HexA2S-HexN-HexA-HexNAc-HexA	0.029618730
HexN-HexA-HexNAc6S-HexA-HexNAc-HexA	0.029618561
HexN-HexA-HexNAc3S-HexA-HexNAc-HexA	0.029573110
HexNAc-HexA-HexN3S-HexA-HexNAc-HexA	0.029473320

Table C.7: Training compound #6 results. Row with correct sequence is italicized.

Sequence	GAGrank score
HexA-HexNS6S-HexA-HexNS6S-HexA-HexNS6S	0.003152165
HexA-HexNS6S-HexA-HexNS6S-HexA-HexNS3S	0.003149186
HexA-HexNS6S-HexA-HexNS3S-HexA-HexNS6S	*0.003149186*
HexA-HexNS3S-HexA-HexNS6S-HexA-HexNS6S	0.003149186
HexA-HexNS6S-HexA-HexNS3S-HexA-HexNS3S	0.003146602
HexA-HexNS3S-HexA-HexNS6S-HexA-HexNS3S	0.003146602
HexA-HexNS3S-HexA-HexNS3S-HexA-HexNS6S	0.003146602
HexA-HexNS6S-HexA-HexNS6S-HexA-HexN3S6S	0.003145289
HexA-HexNS6S-HexA-HexN3S6S-HexA-HexNS6S	0.003145289
HexA-HexN3S6S-HexA-HexNS6S-HexA-HexNS6S	0.003145289

Table C.8: Training compound #7 results. Row with correct sequence is italicized.

Sequence	GAGrank score
HexA-HexNS6S-HexA-HexNS3S6S-HexA2S-HexNS6S	*0.003227609*
HexA-HexNS6S-HexA-HexNS3S6S-HexA2S-HexNS3S	0.003223225
HexA-HexNS3S-HexA-HexNS3S6S-HexA2S-HexNS6S	0.003223225
HexA-HexNS3S-HexA-HexNS3S6S-HexA2S-HexNS3S	0.003219425
HexA-HexNS6S-HexA-HexNS3S6S-HexA2S-HexN3S6S	0.003217493
HexA-HexN3S6S-HexA-HexNS3S6S-HexA2S-HexNS6S	0.003217493
HexA-HexNS3S-HexA-HexNS3S6S-HexA2S-HexN3S6S	0.003214454
HexA-HexN3S6S-HexA-HexNS3S6S-HexA2S-HexNS3S	0.003214454
HexA-HexN3S6S-HexA-HexNS3S6S-HexA2S-HexN3S6S	0.003210481
HexA-HexNS6S-HexA-HexNS6S-HexA2S-HexNS3S6S	0.003204943

Table C.9: Training compound #8 results. Row with correct sequence is italicized.

Sequence	GAGrank score
HexNS-HexA-HexNS-HexA-HexNS-HexA-HexNS-HexA	*0.003375208*
HexN6S-HexA-HexNS-HexA-HexNS-HexA-HexNS-HexA	0.003368827
HexN3S-HexA-HexNS-HexA-HexNS-HexA-HexNS-HexA	0.003362692
HexNS-HexA-HexN6S-HexA-HexNS-HexA-HexNS-HexA	0.003323178
HexNS-HexA-HexNS-HexA-HexNS-HexA-HexN6S-HexA	0.003322108
HexN6S-HexA-HexN6S-HexA-HexNS-HexA-HexNS-HexA	0.003318755
HexN6S-HexA-HexNS-HexA-HexNS-HexA-HexN6S-HexA	0.003317685
HexNS-HexA-HexN3S-HexA-HexNS-HexA-HexNS-HexA	0.003317043
HexNS-HexA-HexNS-HexA-HexNS-HexA-HexN3S-HexA	0.003315973
HexN6S-HexA-HexN3S-HexA-HexNS-HexA-HexNS-HexA	0.003314502

Table C.10: Training compound #9 results. Row with correct sequence is italicized.

Sequence	GAGrank score
HexNAc6S-HexA2S-HexNS6S-HexA2S-HexNS6S-HexA2S-HexNS6S-HexA	0.003412309
HexNAc6S-HexA2S-HexNS6S-HexA2S-HexNS6S-HexA-HexNS3S6S-HexA	0.003412309
HexNAc6S-HexA2S-HexNS6S-HexA2S-HexNS-HexA2S-HexNS3S6S-HexA	0.003412309
HexNAc6S-HexA2S-HexNS6S-HexA-HexNS6S-HexA2S-HexNS3S6S-HexA	0.003412309
HexNAc6S-HexA2S-HexNS6S-HexA-HexNS3S6S-HexA2S-HexNS3S6S-HexA	0.003412309
HexNAc6S-HexA2S-HexNS6S-HexA-HexNS3S6S-HexA-HexNS3S6S-HexA	0.003412309
HexNAc6S-HexA2S-HexNS3S6S-HexA2S-HexNS-HexA2S-HexNS3S6S-HexA	0.003412309
HexNAc6S-HexA2S-HexNS3S6S-HexA2S-HexNS-HexA-HexNS3S6S-HexA	0.003412309
HexNAc6S-HexA2S-HexNS3S6S-HexA-HexNS6S-HexA2S-HexNS3S6S-HexA	0.003412309
HexNAc6S-HexA2S-HexNS3S6S-HexA-HexNS6S-HexA-HexNS3S6S-HexA	0.003412309
HexNAc6S-HexA2S-HexNS3S6S-HexA2S-HexNS-HexA2S-HexNS3S6S-HexA	0.003412309
HexNAc6S-HexA2S-HexNS-HexA2S-HexNS3S6S-HexA-HexNS3S6S-HexA	0.003412309
HexNAc6S-HexA2S-HexNS3S6S-HexA2S-HexNS-HexA2S-HexNS3S6S-HexA	0.003412309
HexNAc6S-HexA2S-HexNS-HexA2S-HexNS3S6S-HexA-HexNS3S6S-HexA	0.003412309
HexNAc6S-HexA2S-HexNS3S6S-HexA2S-HexNS-HexA2S-HexNS3S6S-HexA	0.003412309
HexNAc6S-HexA-HexNS6S-HexA2S-HexNS-HexA2S-HexNS3S6S-HexA	0.003412309
HexNAc6S-HexA-HexNS6S-HexA2S-HexNS3S6S-HexA2S-HexNS3S6S-HexA	0.003412309
HexNAc6S-HexA-HexNS6S-HexA-HexNS3S6S-HexA2S-HexNS3S6S-HexA	0.003412309
HexNAc6S-HexA-HexNS3S6S-HexA2S-HexNS6S-HexA2S-HexNS6S-HexA	*0.003412309*
HexNAc6S-HexA-HexNS3S6S-HexA2S-HexNS6S-HexA-HexNS3S6S-HexA	0.003412309
HexNAc6S-HexA-HexNS3S6S-HexA2S-HexNS-HexA2S-HexNS3S6S-HexA	0.003412309
HexNAc6S-HexA-HexNS3S6S-HexA-HexNS6S-HexA2S-HexNS3S6S-HexA	0.003412309
HexNAc6S-HexA-HexNS3S6S-HexA2S-HexNS6S-HexA-HexNS3S6S-HexA	0.003412309
HexNAc6S-HexA-HexNS-HexA2S-HexNS3S6S-HexA2S-HexNS3S6S-HexA	0.003412309

Table C.11: Training compound #10 results. Row with correct sequence is italicized.

Sequence	GAGrank score
HexNS-HexA-HexNS-HexA-HexNAc-HexA-HexNAc-HexA	*0.009106014*
HexNS-HexA-HexN6S-HexA-HexNAc-HexA-HexNAc-HexA	0.009094131
HexNS-HexA-HexN3S-HexA-HexNAc-HexA-HexNAc-HexA	0.009082706
HexN6S-HexA-HexNS-HexA-HexNAc-HexA-HexNAc-HexA	0.008972119
HexN6S-HexA-HexN6S-HexA-HexNAc-HexA-HexNAc-HexA	0.008963882
HexN6S-HexA-HexN3S-HexA-HexNAc-HexA-HexNAc-HexA	0.008955963
HexN3S-HexA-HexNS-HexA-HexNAc-HexA-HexNAc-HexA	0.008860432
HexN3S-HexA-HexN6S-HexA-HexNAc-HexA-HexNAc-HexA	0.008855701
HexN3S-HexA-HexN3S-HexA-HexNAc-HexA-HexNAc-HexA	0.008851152
HexNS-HexA-HexN-HexA2S-HexNAc-HexA-HexNAc-HexA	0.008812561

Table C.12: Validation compound #1 results. Row with correct sequence is italicized.

Sequence	GAGrank score
HexA-HexNS-HexA2S-HexNS6S	*0.027313000*
HexA2S-HexN-HexA2S-HexNS6S	0.027241385
HexA-HexNS-HexA2S-HexN3S6S	0.027241385
HexA-HexN6S-HexA2S-HexNS6S	0.027241385
HexA2S-HexN-HexA2S-HexN3S6S	0.027191746
HexA-HexN6S-HexA2S-HexN3S6S	0.027191746
HexA-HexN3S-HexA2S-HexNS6S	0.027172529
HexA-HexN3S-HexA2S-HexN3S6S	0.027144018
HexA-HexNS-HexA2S-HexNS3S	0.026174395
HexA2S-HexN-HexA2S-HexNS3S	0.026112302

Table C.13: Validation compound #2 results. Row with correct sequence is italicized.

Sequence	GAGrank score
HexA-HexNS6S-HexA-HexNS6S-HexA2S-HexNS6S	*0.005042149*
HexA-HexNS6S-HexA-HexNS3S-HexA2S-HexNS6S	0.005039053
HexA-HexNS3S-HexA-HexNS6S-HexA2S-HexNS6S	0.005039053
HexA-HexNS3S-HexA-HexNS3S-HexA2S-HexNS6S	0.005036369
HexA-HexN3S6S-HexA-HexNS6S-HexA2S-HexNS6S	0.005035004
HexA-HexN3S6S-HexA-HexNS3S-HexA2S-HexNS6S	0.005032858
HexA-HexNS6S-HexA-HexNS6S-HexA2S-HexNS3S	0.004997753
HexA-HexNS6S-HexA-HexN3S6S-HexA2S-HexNS6S	0.004995329
HexA-HexNS6S-HexA-HexNS3S-HexA2S-HexNS3S	0.004995069
HexA-HexNS3S-HexA-HexNS6S-HexA2S-HexNS3S	0.004995069

Table C.14: Validation compound #3 results. Row with correct sequence is italicized.

Sequence	GAGrank score
HexA-HexNS6S-HexA2S-HexNS6S-HexA2S-HexNS6S	0.004176479
HexA2S-HexNS-HexA2S-HexNS6S-HexA2S-HexNS6S	0.004176474
HexA-HexNS6S-HexA2S-HexNS6S-HexA2S-HexNS3S	0.004172096
HexA-HexNS6S-HexA2S-HexNS3S-HexA2S-HexNS6S	*0.004172096*
HexA2S-HexNS-HexA2S-HexNS6S-HexA2S-HexNS3S	0.004172090
HexA2S-HexNS-HexA2S-HexNS3S-HexA2S-HexNS6S	0.004172090
HexA-HexNS6S-HexA2S-HexNS3S-HexA2S-HexNS3S	0.004168295
HexA2S-HexNS-HexA2S-HexNS3S-HexA2S-HexNS3S	0.004168290
HexA-HexNS6S-HexA2S-HexNS6S-HexA2S-HexN3S6S	0.004166363
HexA-HexN3S6S-HexA2S-HexNS6S-HexA2S-HexNS6S	0.004166363

Table C.15: Mixture compound #1 100:0 results. Row(s) with correct sequence(s) is italicized.

Sequence	GAGrank score
HexA-HexNS6S-HexA-HexNS3S6S-HexA-HexNS6S	*0.002950081*
HexA-HexNS3S-HexA-HexNS3S6S-HexA-HexNS6S	0.002945489
HexA-HexNS6S-HexA-HexNS3S6S-HexA-HexN3S6S	0.002940470
HexA-HexN3S6S-HexA-HexNS3S6S-HexA-HexNS6S	0.002940470
HexA-HexNS3S-HexA-HexNS3S6S-HexA-HexN3S6S	0.002938263
HexA-HexN3S6S-HexA-HexNS3S6S-HexA-HexN3S6S	0.002935852
HexA-HexNS6S-HexA2S-HexNS6S-HexA-HexNS6S	0.002931145
HexA-HexNS6S-HexA2S-HexNS3S-HexA-HexNS6S	*0.002926553*
HexA-HexNS3S-HexA2S-HexNS6S-HexA-HexNS6S	0.002926553
HexA-HexNS3S-HexA2S-HexNS3S-HexA-HexNS6S	0.002923101

Table C.16: Mixture compound #1 90:10 results. Row(s) with correct sequence(s) is italicized.

Sequence	GAGrank score
HexA-HexNS6S-HexA-HexNS3S6S-HexA-HexNS6S	*0.002753264*
HexA-HexNS3S-HexA-HexNS3S6S-HexA-HexNS6S	0.002748671
HexA-HexNS6S-HexA-HexNS3S6S-HexA-HexN3S6S	0.002743652
HexA-HexN3S6S-HexA-HexNS3S6S-HexA-HexNS6S	0.002743652
HexA-HexNS3S-HexA-HexNS3S6S-HexA-HexN3S6S	0.002741446
HexA-HexN3S6S-HexA-HexNS3S6S-HexA-HexN3S6S	0.002739035
HexA-HexNS6S-HexA2S-HexNS6S-HexA-HexNS6S	0.002724934
HexA2S-HexNS-HexA-HexNS3S6S-HexA-HexNS6S	0.002724480
HexA-HexNS6S-HexA-HexNS3S6S-HexA-HexNS3S	0.002720716
HexA-HexNS6S-HexA2S-HexNS3S-HexA-HexNS6S	*0.002720341*

Table C.17: Mixture compound #1 70:30 results. Row(s) with correct sequence(s) is italicized.

Sequence	GAGrank score
HexA-HexNS6S-HexA-HexNS3S6S-HexA-HexNS6S	*0.002234407*
HexA-HexNS3S-HexA-HexNS3S6S-HexA-HexNS6S	0.002229814
HexA-HexNS6S-HexA-HexNS3S6S-HexA-HexN3S6S	0.002224795
HexA-HexN3S6S-HexA-HexNS3S6S-HexA-HexNS6S	0.002224795
HexA-HexNS3S-HexA-HexNS3S6S-HexA-HexN3S6S	0.002222589
HexA-HexN3S6S-HexA-HexNS3S6S-HexA-HexN3S6S	0.002220177
HexA-HexNS6S-HexA2S-HexNS6S-HexA-HexNS6S	0.002210470
HexA-HexNS6S-HexA2S-HexNS3S-HexA-HexNS6S	*0.002205878*
HexA-HexNS3S-HexA2S-HexNS6S-HexA-HexNS6S	0.002205878
HexA-HexNS6S-HexA-HexNS3S6S-HexA-HexNS3S	0.002204972

Table C.18: Mixture compound #1 50:50 results. Row(s) with correct sequence(s) is italicized.

Sequence	GAGrank score
HexA-HexNS6S-HexA2S-HexNS6S-HexA-HexNS6S	0.003179077
HexA-HexNS6S-HexA2S-HexNS3S-HexA-HexNS6S	*0.003174485*
HexA-HexNS3S-HexA2S-HexNS6S-HexA-HexNS6S	0.003174485
HexA-HexNS3S-HexA2S-HexNS3S-HexA-HexNS6S	0.003171033
HexA-HexNS6S-HexA2S-HexNS6S-HexA-HexN3S6S	0.003169466
HexA-HexNS6S-HexA2S-HexN3S6S-HexA-HexNS6S	0.003169466
HexA-HexN3S6S-HexA2S-HexNS6S-HexA-HexNS6S	0.003169466
HexA-HexNS6S-HexA2S-HexNS3S-HexA-HexN3S6S	0.003167259
HexA-HexNS3S-HexA2S-HexNS6S-HexA-HexN3S6S	0.003167259
HexA-HexNS3S-HexA2S-HexN3S6S-HexA-HexNS6S	0.003167259

Table C.19: Mixture compound #1 30:70 results. Row(s) with correct sequence(s) is italicized.

Sequence	GAGrank score
HexA-HexNS6S-HexA2S-HexNS6S-HexA-HexNS6S	0.002927872
HexA-HexNS6S-HexA2S-HexNS6S-HexA-HexNS3S	0.002923280
HexA-HexNS6S-HexA2S-HexNS3S-HexA-HexNS6S	*0.002923280*
HexA-HexNS3S-HexA2S-HexNS6S-HexA-HexNS6S	0.002923280
HexA-HexNS6S-HexA2S-HexNS3S-HexA-HexNS3S	0.002919827
HexA-HexNS3S-HexA2S-HexNS6S-HexA-HexNS3S	0.002919827
HexA-HexNS3S-HexA2S-HexNS3S-HexA-HexNS6S	0.002919827
HexA-HexNS6S-HexA2S-HexNS6S-HexA-HexN3S6S	0.002918261
HexA-HexNS6S-HexA2S-HexN3S6S-HexA-HexNS6S	0.002918261
HexA-HexN3S6S-HexA2S-HexNS6S-HexA-HexNS6S	0.002918261

Table C.20: Mixture compound #1 10:90 results. Row(s) with correct sequence(s) is italicized.

Sequence	GAGrank score
HexA-HexNS6S-HexA2S-HexNS6S-HexA-HexNS6S	0.003106300
HexA-HexNS6S-HexA2S-HexNS6S-HexA-HexNS3S	0.003101708
HexA-HexNS6S-HexA2S-HexNS3S-HexA-HexNS6S	*0.003101708*
HexA-HexNS3S-HexA2S-HexNS6S-HexA-HexNS6S	0.003101708
HexA-HexNS6S-HexA2S-HexNS3S-HexA-HexNS3S	0.003098255
HexA-HexNS3S-HexA2S-HexNS6S-HexA-HexNS3S	0.003098255
HexA-HexNS3S-HexA2S-HexNS3S-HexA-HexNS6S	0.003098255
HexA-HexNS6S-HexA2S-HexNS6S-HexA-HexN3S6S	0.003096689
HexA-HexNS6S-HexA2S-HexN3S6S-HexA-HexNS6S	0.003096689
HexA-HexN3S6S-HexA2S-HexNS6S-HexA-HexNS6S	0.003096689

Table C.21: Mixture compound #1 0:100 results. Row(s) with correct sequence(s) is italicized.

Sequence	GAGrank score
HexA-HexNS6S-HexA2S-HexNS6S-HexA-HexNS6S	0.003356457
HexA-HexNS6S-HexA2S-HexNS6S-HexA-HexNS3S	0.003351864
HexA-HexNS6S-HexA2S-HexNS3S-HexA-HexNS6S	*0.003351864*
HexA-HexNS3S-HexA2S-HexNS6S-HexA-HexNS6S	0.003351864
HexA-HexNS6S-HexA2S-HexNS3S-HexA-HexNS3S	0.003348412
HexA-HexNS3S-HexA2S-HexNS6S-HexA-HexNS3S	0.003348412
HexA-HexNS3S-HexA2S-HexNS3S-HexA-HexNS6S	0.003348412
HexA-HexNS6S-HexA2S-HexNS6S-HexA-HexN3S6S	0.003346845
HexA-HexNS6S-HexA2S-HexN3S6S-HexA-HexNS6S	0.003346845
HexA-HexN3S6S-HexA2S-HexNS6S-HexA-HexNS6S	0.003346845

Table C.22: Mixture compound #2 100:0 results. Row(s) with correct sequence(s) is italicized.

Sequence	GAGrank score
HexA-HexNAc6S-HexA-HexNAc6S	*0.050216157*
HexA-HexNAc3S-HexA-HexNAc6S	0.050113371
HexA-HexNAc6S-HexA-HexNAc3S	0.049088079
HexA-HexNAc3S-HexA-HexNAc3S	0.049010809
HexA2S-HexNAc-HexA-HexNAc6S	0.044482264
HexA2S-HexNAc-HexA-HexNAc3S	0.043217273
HexA-HexNAc6S-HexA2S-HexNAc	0.042408065
HexA-HexNAc-HexA2S-HexNAc6S	*0.042394732*
HexA-HexNAc3S-HexA2S-HexNAc	0.042305279
HexA-HexNAc-HexA2S-HexNAc3S	0.041070433

Table C.23: Mixture compound #2 90:10 results. Row(s) with correct sequence(s) is italicized.

Sequence	GAGrank score
HexA-HexNAc6S-HexA-HexNAc6S	*0.050206896*
HexA-HexNAc3S-HexA-HexNAc6S	0.050108197
HexA-HexNAc6S-HexA-HexNAc3S	0.049038267
HexA-HexNAc3S-HexA-HexNAc3S	0.048964070
HexA2S-HexNAc-HexA-HexNAc6S	0.044165358
HexA-HexNAc-HexA2S-HexNAc6S	*0.043328393*
HexA2S-HexNAc-HexA-HexNAc3S	0.042844929
HexA-HexNAc6S-HexA2S-HexNAc	0.042038936
HexA-HexNAc-HexA2S-HexNAc3S	0.041983674
HexA-HexNAc3S-HexA2S-HexNAc	0.041940237

Table C.24: Mixture compound #2 70:30 results. Row(s) with correct sequence(s) is italicized.

Sequence	GAGrank score
HexA-HexNAc6S-HexA-HexNAc6S	*0.049219907*
HexA-HexNAc3S-HexA-HexNAc6S	0.049121208
HexA-HexNAc6S-HexA-HexNAc3S	0.048072594
HexA-HexNAc3S-HexA-HexNAc3S	0.047998397
HexA-HexNAc-HexA2S-HexNAc6S	*0.044810770*
HexA-HexNAc-HexA2S-HexNAc3S	0.043556734
HexA2S-HexNAc-HexA-HexNAc6S	0.043326820
HexA2S-HexNAc-HexA-HexNAc3S	0.042031565
HexA-HexNAc6S-HexA2S-HexNAc	0.041193679
HexA-HexNAc3S-HexA2S-HexNAc	0.041094980

Table C.25: Mixture compound #2 50:50 results. Row(s) with correct sequence(s) is italicized.

Sequence	GAGrank score
HexA-HexNAc-HexA2S-HexNAc6S	*0.041595545*
HexA-HexNAc6S-HexA-HexNAc6S	*0.041053720*
HexA-HexNAc3S-HexA-HexNAc6S	0.040958795
HexA-HexNAc-HexA2S-HexNAc3S	0.040662793
HexA-HexNAc6S-HexA-HexNAc3S	0.040106248
HexA-HexNAc3S-HexA-HexNAc3S	0.040034887
HexA-HexNAc-HexA-HexNAc3S6S	0.036556982
HexA2S-HexNAc-HexA-HexNAc6S	0.036452322
HexA2S-HexNAc-HexA-HexNAc3S	0.035393278
HexA-HexNAc6S-HexA2S-HexNAc	0.034580137

Table C.26: Mixture compound #2 30:70 results. Row(s) with correct sequence(s) is italicized.

Sequence	GAGrank score
HexA-HexNAc-HexA2S-HexNAc6S	*0.042476891*
HexA-HexNAc-HexA2S-HexNAc3S	0.041595135
HexA-HexNAc6S-HexA-HexNAc6S	*0.040232800*
HexA-HexNAc3S-HexA-HexNAc6S	0.040137875
HexA-HexNAc6S-HexA-HexNAc3S	0.039302779
HexA-HexNAc3S-HexA-HexNAc3S	0.039231418
HexA-HexNAc-HexA-HexNAc3S6S	0.037743468
HexA2S-HexNAc-HexA-HexNAc6S	0.034790011
HexA-HexNAc6S-HexA2S-HexNAc	0.033899028
HexA-HexNAc3S-HexA2S-HexNAc	0.033804104

Table C.27: Mixture compound #2 10:90 results. Row(s) with correct sequence(s) is italicized.

Sequence	GAGrank score
HexA-HexNAc-HexA2S-HexNAc6S	*0.045719214*
HexA-HexNAc-HexA2S-HexNAc3S	0.044614502
HexA-HexNAc-HexA-HexNAc3S6S	0.041820284
HexA-HexNAc6S-HexA-HexNAc6S	*0.041002384*
HexA-HexNAc3S-HexA-HexNAc6S	0.040907459
HexA-HexNAc6S-HexA-HexNAc3S	0.039772429
HexA-HexNAc3S-HexA-HexNAc3S	0.039701068
HexA2S-HexNAc-HexA-HexNAc6S	0.036061804
HexA-HexNAc6S-HexA2S-HexNAc	0.034878209
HexA-HexNAc3S-HexA2S-HexNAc	0.034783284

Table C.28: Mixture compound #2 0:100 results. Row(s) with correct sequence(s) is italicized.

Sequence	GAGrank score
HexA-HexNAc-HexA2S-HexNAc6S	*0.058691828*
HexA-HexNAc-HexA2S-HexNAc3S	0.057379903
HexA-HexNAc-HexA-HexNAc3S6S	0.055061707
HexA-HexNAc3S-HexA-HexNAc6S	0.050889993
HexA-HexNAc6S-HexA-HexNAc6S	*0.049617336*
HexA-HexNAc3S-HexA-HexNAc3S	0.049419760
HexA-HexNAc6S-HexA-HexNAc3S	0.048078461
HexA-HexNAc3S-HexA2S-HexNAc	0.045131311
HexA-HexNAc6S-HexA2S-HexNAc	0.043674420
HexA-HexNAc3S6S-HexA-HexNAc	0.043674420

References

[1] Subramanian, S. V., Fitzgerald, M. L. & Bernfield, M. Regulated Shedding of Syndecan-1 and -4 Ectodomains by Thrombin and Growth Factor Receptor Activation. *Journal of Biological Chemistry* **272**, 14713–14720 (1997).

[2] Ruiz, X. D. *et al.* Syndecan-2 Is a Novel Target of Insulin-Like Growth Factor Binding Protein-3 and Is Over-Expressed in Fibrosis. *PLOS One* **7**, e43049 (2012).

[3] Wang, H., Jin, H., Beauvais, D. M. & Rapraeger, A. C. Cytoplasmic Domain Interactions of Syndecan-1 and Syndecan-4 with $a6\beta4$ Integrin Mediate Human Epidermal Growth Factor Receptor (HER1 and HER2)-dependent Motility and Survival. *Journal of Biological Chemistry* **289**, 30318–30332 (2014).

[4] Wang, H., Jin, H. & Rapraeger, A. C. Syndecan-1 and Syndecan-4 Capture Epidermal Growth Factor Receptor Family Members and the $a3\beta1$ Integrin Via Binding Sites in Their Ectodomains NOVEL SYNSTATINS PREVENT KINASE CAPTURE AND INHIBIT $a6\beta4$-INTEGRIN-DEPENDENT EPITHELIAL CELL MOTILITY. *Journal of Biological Chemistry* **290**, 26103–26113 (2015).

[5] Elfenbein, A. & Simons, M. Syndecan-4 signaling at a glance. *Journal of Cell Science* **126**, 3799–3804 (2013).

[6] Okina, E., Grossi, A., Gopal, S., Multhaupt, H. A. B. & Couchman, J. R. Alpha-actinin interactions with syndecan-4 are integral to fibroblast–matrix adhesion and regulate cytoskeletal architecture. *The International Journal of Biochemistry & Cell Biology* **44**, 2161–2174 (2012).

[7] Gopal, S., Multhaupt, H. A. B., Pocock, R. & Couchman, J. R. Cell-extracellular matrix and cell-cell adhesion are linked by syndecan-4. *Matrix Biology* **60-61**, 57–69 (2017).

[8] Mitsou, I., Multhaupt, H. A. B. & Couchman, J. R. Proteoglycans, ion channels and cell–matrix adhesion. *Biochemical Journal* **474**, 1965–1979 (2017).

[9] Altemeier, W. A., Schlesinger, S. Y., Buell, C. A., Parks, W. C. & Chen, P. Syndecan-1 controls cell migration by activating Rap1 to regulate focal adhesion disassembly. *Journal of Cell Science* **125**, 5188–5195 (2012).

[10] Polisetti, N., Zenkel, M., Menzel-Severing, J., Kruse, F. E. & Schlötzer-Schrehardt, U. Cell Adhesion Molecules and Stem Cell-Niche-Interactions in the Limbal Stem Cell Niche. *Stem Cells* **34**, 203–219 (2016).

[11] Cain, S. A., Mularczyk, E. J., Singh, M., Massam-Wu, T. & Kielty, C. M. ADAMTS-10 and -6 differentially regulate cell-cell junctions and focal adhesions. *Scientific Reports* **6** (2016).

[12] Sun, M. *et al.* RKIP and HMGA2 regulate breast tumor survival and metastasis through lysyl oxidase and syndecan-2. *Oncogene* **33**, 3528–3537 (2014).

[13] Song, H. H. & Filmus, J. The role of glypicans in mammalian development. *Biochimica et Biophysica Acta* **1573**, 241–246 (2002).

[14] Kobayashi, K., Ding, G., Nishikawa, S.-I. & Kataoka, H. Role of Etv2-positive cells in the remodeling morphogenesis during vascular development. *Genes to Cells* **18**, 704–721 (2013).

[15] Blair, S. S. Cell Signaling: Wingless and Glypicans Together Again. *Current Biology* **15**, R92–R94 (2005).

[16] Fico, A., Maina, F. & Dono, R. Fine-tuning of cell signaling by glypicans. *Cellular and Molecular Life Sciences* **68**, 923–929 (2011).

[17] Deguchi, Y. *et al.* Internalization of basic fibroblast growth factor at the mouse blood–brain barrier involves perlecan, a heparan sulfate proteoglycan. *Journal of Neurochemistry* **83**, 381–389 (2002).

[18] Parham, C., Auckland, L., Rachwal, J., Clarke, D. & Bix, G. Perlecan Domain V Inhibits Amyloid-β Induced Brain Endothelial Cell Toxicity and Restores Angiogenic Function. *Journal of Alzheimer's Disease* **38**, 415–423 (2014).

[19] Godfrey, E. W., Roe, J. & Heathcote, R. D. Overexpression of Agrin Isoforms in Xenopus Embryos Alters the Distribution of Synaptic Acetylcholine Receptors during Development of the Neuromuscular Junction. *Developmental Biology* **205**, 22–32 (1999).

[20] Anselmo, A. *et al.* Identification of a novel agrin-dependent pathway in cell signaling and adhesion within the erythroid niche. *Cell Death and Differentiation* **23**, 1322–1330 (2016).

[21] Neame, P. J., Kay, C. J., McQuillan, D. J., Beales, M. P. & Hassell, J. R. Independent modulation of collagen fibrillogenesis by decorin and lumican. *Cellular and Molecular Life Sciences* **57**, 859–863 (2000).

[22] Rühland, C. *et al.* The glycosaminoglycan chain of decorin plays an important role in collagen fibril formation at the early stages of fibrillogenesis. *The FEBS Journal* **274**, 4246–4255 (2007).

[23] Zhang, G. *et al.* Genetic Evidence for the Coordinated Regulation of Collagen Fibrillogenesis in the Cornea by Decorin and Biglycan. *Journal of Biological Chemistry* **284**, 8888–8897 (2009).

[24] Reese, S. P., Underwood, C. J. & Weiss, J. A. Effects of decorin proteoglycan on fibrillogenesis, ultrastructure, and mechanics of type I collagen gels. *Matrix Biology* **32**, 414–423 (2013).

[25] Wallace, J. M. *et al.* The mechanical phenotype of biglycan-deficient mice is bone- and gender-specific. *Bone* **39**, 106–116 (2006).

[26] Landolt, R. M., Vaughan, L., Winterhalter, K. H. & Zimmermann, D. R. Versican is selectively expressed in embryonic tissues that act as barriers to neural crest cell migration and axon outgrowth. *Development* **121**, 2303–2312 (1995).

[27] Inai, K., Burnside, J. L., Hoffman, S., Toole, B. P. & Sugi, Y. BMP-2 Induces Versican and Hyaluronan That Contribute to Post-EMT AV Cushion Cell Migration. *PLOS One* **8**, e77593 (2013).

[28] Onken, J. *et al.* Versican isoform V1 regulates proliferation and migration in high-grade gliomas. *Journal of Neuro-Oncology* **120**, 73–83 (2014).

[29] Cichon, S. *et al.* Genome-wide Association Study Identifies Genetic Variation in Neurocan as a Susceptibility Factor for Bipolar Disorder. *The American Journal of Human Genetics* **88**, 372–381 (2011).

[30] Sandy, J. D., Plaas, A. H. & Koob, T. J. Pathways of aggrecan processing in joint tissues. Implications for disease mechanism and monitoring. *Acta Orthopaedica Scandinavica* **266**, 26–32 (1995).

[31] Lohmander, L. S., Ionescu, M., Jugessur, H. & Poole, A. R. Changes in joint cartilage aggrecan after knee injury and in osteoarthritis. *Arthritis & Rheumatism* **42**, 534–544 (1999).

[32] Nutt, C. L., Matthews, R. T. & Hockfield, S. Glial Tumor Invasion: A Role for the Upregulation and Cleavage of BEHAB/Brevican. *The Neuroscientist* **7**, 113–122 (2001).

[33] Lu, R. *et al.* The role of brevican in glioma: Promoting tumor cell motility in vitro and in vivo. *BMC Cancer* **12**, 607 (2012).

[34] Dwyer, C. A., Bi, W. L., Viapiano, M. S. & Matthews, R. T. Brevican knockdown reduces late-stage glioma tumor aggressiveness. *Journal of Neuro-Oncology* **120**, 63–72 (2014).

[35] Ezura, Y., Chakravarti, S., Oldberg, A., Chervoneva, I. & Birk, D. E. Differential Expression of Lumican and Fibromodulin Regulate Collagen Fibrillogenesis in Developing Mouse Tendons. *The Journal of Cell Biology* **151**, 779–788 (2000).

[36] Kao, W. W.-Y. & Liu, C.-Y. Roles of lumican and keratocan on corneal transparency. *Glycoconjugate Journal* **19**, 275–285 (2002).

[37] Pellegata, N. S. *et al.* Mutations in *KERA*, encoding keratocan, cause cornea plana. *Nature Genetics* **25**, 91–95 (2000).

[38] Lehmann, O. J. *et al.* A Novel Keratocan Mutation Causing Autosomal Recessive Cornea Plana. *Investigative Ophthalmology & Visual Science* **42**, 3118–3122 (2001).

[39] Afratis, N. *et al.* Glycosaminoglycans: Key players in cancer cell biology and treatment. *The FEBS Journal* **279**, 1177–1197 (2012).

[40] Pinho, S. S. & Reis, C. A. Glycosylation in cancer: Mechanisms and clinical implications. *Nature Reviews Cancer* **15**, 540–555 (2015).

[41] Sethi, M. K., Hancock, W. S. & Fanayan, S. Identifying *N*-Glycan Biomarkers in Colorectal Cancer by Mass Spectrometry. *Accounts of Chemical Research* **49**, 2099–2106 (2016).

[42] Carvalho, S., Reis, C. A. & Pinho, S. S. Cadherins Glycans in Cancer: Sweet Players in a Bitter Process. *Trends in Cancer* **2**, 519–531 (2016).

[43] Chen, H. *et al.* Mass spectrometric profiling reveals association of *N*-glycan patterns with epithelial ovarian cancer progression. *Tumor Biology* **39** (2017).

[44] Veillon, L., Fakih, C., Abou-El-Hassan, H., Kobeissy, F. & Mechref, Y. Glycosylation Changes in Brain Cancer. *ACS Chemical Neuroscience* **9**, 51–72 (2018).

[45] Aoki-Kinoshita, K. *et al.* GlyTouCan 1.0 – The international glycan structure repository. *Nucleic Acids Research* **44**, D1237–D1242 (2016).

[46] Tiemeyer, M. *et al.* GlyTouCan: An accessible glycan structure repository. *Glycobiology* **27**, 915–919 (2017).

[47] Campbell, M. P. *et al.* UniCarbKB: Building a knowledge platform for glycoproteomics. *Nucleic Acids Research* **42**, D215–D221 (2014).

[48] Ranzinger, R., Herget, S., von der Lieth, C.-W. & Frank, M. GlycomeDB—a unified database for carbohydrate structures. *Nucleic Acids Research* **39**, D373–D376 (2011).

[49] Toukach, P. V. Bacterial Carbohydrate Structure Database 3: Principles and Realization. *Journal of Chemical Information and Modeling* **51**, 159–170 (2011).

[50] Okuda, S., Nakao, H. & Kawasaki, T. GlycoEpitope: Database for Carbohydrate Antigen and Antibody. In Taniguchi, N., Endo, T., Hart, G. W., Seeberger, P. H. & Wong, C.-H. (eds.) *Glycoscience: Biology and Medicine*, 267–273 (Springer Japan, Tokyo, 2015).

[51] Cooper, C. A., Gasteiger, E. & Packer, N. H. GlycoMod – A software tool for determining glycosylation compositions from mass spectrometric data. *Proteomics* **1**, 340–349 (2001).

[52] Bateman, A. *et al.* UniProt: The universal protein knowledgebase. *Nucleic Acids Research* **45**, D158–D169 (2017).

[53] Joshi, H. J. *et al.* Development of a mass fingerprinting tool for automated interpretation of oligosaccharide fragmentation data. *Proteomics* **4**, 1650–1664 (2004).

[54] Cooper, C. A., Harrison, M. J., Wilkins, M. R. & Packer, N. H. GlycoSuiteDB: A new curated relational database of glycoprotein glycan structures and their biological sources. *Nucleic Acids Research* **29**, 332–335 (2001).

[55] Lohmann, K. K. & von der Lieth, C.-W. GlycoFragment and GlycoSearchMS: Web tools to support the interpretation of mass spectra of complex carbohydrates. *Nucleic Acids Research* **32**, W261–W266 (2004).

[56] Loß, A. *et al.* SWEET-DB: An attempt to create annotated data collections for carbohydrates. *Nucleic Acids Research* **30**, 405–408 (2002).

[57] Ren, J. M., Rejtar, T., Li, L. & Karger, B. L. *N*-Glycan Structure Annotation of Glycopeptides Using a Linearized Glycan Structure Database (GlyDB). *Journal of Proteome Research* **6**, 3162–3173 (2007).

[58] Woodin, C. L. *et al.* GlycoPep Grader: A Web-Based Utility for Assigning the Composition of *N*-Linked Glycopeptides. *Analytical Chemistry* **84**, 4821–4829 (2012).

[59] Zhu, Z., Hua, D., Clark, D. F., Go, E. P. & Desaire, H. GlycoPep Detector: A Tool for Assigning Mass Spectrometry Data of *N*-Linked Glycopeptides on the Basis of Their Electron Transfer Dissociation Spectra. *Analytical Chemistry* **85**, 5023–5032 (2013).

[60] He, L., Xin, L., Shan, B., Lajoie, G. A. & Ma, B. GlycoMaster DB: Software To Assist the Automated Identification of *N*-Linked Glycopeptides by Tandem Mass Spectrometry. *Journal of Proteome Research* **13**, 3881–3895 (2014).

[61] Mayampurath, A. *et al.* Computational Framework for Identification of Intact Glycopeptides in Complex Samples. *Analytical Chemistry* **86**, 453–463 (2014).

[62] Zeng, W.-F. *et al.* pGlyco: A pipeline for the identification of intact *N*-glycopeptides by using HCD- and CID-MS/MS and MS^3. *Scientific Reports* **6** (2016).

[63] Gaucher, S. P., Morrow, J. & Leary, J. A. STAT: A Saccharide Topology Analysis Tool Used in Combination with Tandem Mass Spectrometry. *Analytical Chemistry* **72**, 2331–2336 (2000).

[64] Ethier, M., Saba, J. A., Ens, W., Standing, K. G. & Perreault, H. Automated structural assignment of derivatized complex *N*-linked oligosaccharides from tandem mass spectra. *Rapid Communications in Mass Spectrometry* **16**, 1743–1754 (2002).

[65] Ethier, M. *et al.* Application of the StrOligo algorithm for the automated structure assignment of complex *N*-linked glycans from glycoproteins using tandem mass spectrometry. *Rapid Communications in Mass Spectrometry* **17**, 2713–2720 (2003).

[66] Tang, H., Mechref, Y. & Novotny, M. V. Automated interpretation of MS/MS spectra of oligosaccharides. *Bioinformatics* **21**, i431–i439 (2005).

[67] Peltoniemi, H., Joenväärä, S. & Renkonen, R. *De Novo* glycan structure search with the CID MS/MS spectra of native *N*-glycopeptides. *Glycobiology* **19**, 707–714 (2009).

[68] Böcker, S., Kehr, B. & Rasche, F. Determination of Glycan Structure from Tandem Mass Spectra. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **8**, 976–986 (2011).

[69] Dong, L. *et al.* An Accurate *De Novo* Algorithm for Glycan Topology Determination from Mass Spectra. *IEEE/ACM Trans. Comput. Biol. Bioinformatics* **12**, 568–578 (2015).

[70] Sun, W. *et al.* An Effective Approach for Glycan Structure *De Novo* Sequencing From HCD Spectra. *IEEE Transactions on NanoBioscience* **15**, 177–184 (2016).

[71] Hong, P. *et al.* GlycoDeNovo – an Efficient Algorithm for Accurate *de Novo* Glycan Topology Reconstruction from Tandem Mass Spectra. *Journal of the American Society for Mass Spectrometry* **28**, 2288–2301 (2017).

[72] Saad, O. M. & Leary, J. A. Heparin Sequencing Using Enzymatic Digestion and ESI-MS^n with HOST: A Heparin/HS Oligosaccharide Sequencing Tool. https://pubs.acs.org/doi/abs/10.1021/ac050793d (2005).

[73] Tissot, B. *et al.* Software Tool for the Structural Determination of Glycosaminoglycans by Mass Spectrometry. *Analytical Chemistry* **80**, 9204–9212 (2008).

[74] Ceroni, A. *et al.* GlycoWorkbench: A Tool for the Computer-Assisted Annotation of Mass Spectra of Glycans. *Journal of Proteome Research* **7**, 1650–1659 (2008).

[75] Spencer, J. L., Bernanke, J. A., Buczek-Thomas, J. A. & Nugent, M. A. A Computational Approach for Deciphering the Organization of Glycosaminoglycans. *PLOS One* **5**, e9389 (2010).

[76] Hu, H. *et al.* A Computational Framework for Heparan Sulfate Sequencing Using High-resolution Tandem Mass Spectra. *Molecular & Cellular Proteomics* **13**, 2490–2502 (2014).

[77] Chiu, Y., Huang, R., Orlando, R. & Sharp, J. S. GAG-ID: Heparan Sulfate (HS) and Heparin Glycosaminoglycan High-Throughput Identification Software. *Molecular & Cellular Proteomics* **14**, 1720–1730 (2015).

[78] Chiu, Y., Schliekelman, P., Orlando, R. & Sharp, J. S. A Multivariate Mixture Model to Estimate the Accuracy of Glycosaminoglycan Identifications Made by Tandem Mass Spectrometry (MS/MS) and Database Search. *Molecular & Cellular Proteomics* **16**, 255–264 (2017).

[79] Duan, J. & Amster, I. J. An Automated, High-Throughput Method for Interpreting the Tandem Mass Spectra of Glycosaminoglycans. *Journal of The American Society for Mass Spectrometry* **29**, 1802–1811 (2018).

[80] Wolff, J. J., Chi, L., Linhardt, R. J. & Amster, I. J. Distinguishing Glucuronic from Iduronic Acid in Glycosaminoglycan Tetrasaccharides by Using Electron Detachment Dissociation. *Analytical Chemistry* **79**, 2015–2022 (2007).

[81] Kailemia, M. J. *et al.* Differentiating Chondroitin Sulfate Glycosaminoglycans Using Collision-Induced Dissociation; Uronic Acid Cross-Ring Diagnostic Fragments in a Single Stage of Tandem Mass Spectrometry. *European Journal of Mass Spectrometry* **21**, 275–285 (2015).

[82] Agyekum, I., Zong, C., Boons, G.-J. & Amster, I. J. Single Stage Tandem Mass Spectrometry Assignment of the C-5 Uronic Acid Stereochemistry in Heparan Sulfate Tetrasaccharides using Electron Detachment Dissociation. *Journal of The American Society for Mass Spectrometry* **28**, 1741–1750 (2017).

[83] Bishop, J. R., Schuksz, M. & Esko, J. D. Heparan sulphate proteoglycans fine-tune mammalian physiology. *Nature* **446**, 1030–1037 (2007).

[84] Lindahl, U. & Li, J.-p. Chapter 3 Interactions Between Heparan Sulfate and Proteins—Design and Functional Implications. In *International Review of Cell and Molecular Biology*, vol. 276 of *International Review of Cell and Molecular Biology*, 105–159 (Academic Press, 2009).

[85] Fuster, M. M. & Esko, J. D. The sweet and sour of cancer: Glycans as novel therapeutic targets. *Nature Reviews Cancer* **5**, 526–542 (2005).

[86] Hirano, K. *et al.* Ablation of Keratan Sulfate Accelerates Early Phase Pathogenesis of ALS. *PLOS One* **8**, e66969 (2013).

[87] Perrimon, N. & Bernfield, M. Specificities of heparan sulphate proteoglycans in developmental processes. *Nature* **404**, 725–728 (2000).

[88] Wolff, J. J. *et al.* Negative Electron Transfer Dissociation of Glycosaminoglycans. https://pubs.acs.org/doi/abs/10.1021/ac100554a (2010).

[89] Huang, Y. *et al. De Novo* sequencing of Heparan Sulfate Oligosaccharides by Electron-Activated Dissociation. *Analytical Chemistry* **85**, 11979–11986 (2013).

[90] Horn, D. M., Zubarev, R. A. & McLafferty, F. W. Automated reduction and interpretation of high resolution electrospray mass spectra of large molecules. *Journal of the American Society for Mass Spectrometry* **11**, 320–332 (2000).

[91] Jaitly, N. *et al.* Decon2LS: An open-source software package for automated processing and visualization of high resolution mass spectrometry data. *BMC Bioinformatics* **10**, 87 (2009).

[92] Liu, X. *et al.* Deconvolution and Database Search of Complex Tandem Mass Spectra of Intact Proteins. *Molecular & Cellular Proteomics* **9**, 2772–2782 (2010).

[93] Yu, Y. *et al.* Sequencing the Dermatan Sulfate Chain of Decorin. *Journal of the American Chemical Society* **139**, 16986–16995 (2017).

[94] Deutsch, E. mzML: A single, unifying data format for mass spectrometer output. *Proteomics* **8**, 2776–2777 (2008).

[95] Chambers, M. C. *et al.* A cross-platform toolkit for mass spectrometry and proteomics. *Nature Biotechnology* **30**, 918–920 (2012).

[96] Bald, T. *et al.* pymzML–Python module for high-throughput bioinformatics on mass spectrometry data. *Bioinformatics* **28**, 1052–1053 (2012).

[97] Kaur, P. & O'Connor, P. B. Algorithms for Automatic Interpretation of High Resolution Mass Spectra. *Journal of the American Society for Mass Spectrometry* **17**, 459–468 (2006).

[98] Dittwald, P., Claesen, J., Burzykowski, T., Valkenborg, D. & Gambin, A. BRAIN: A universal tool for high-throughput calculations of the isotopic distribution for mass spectrometry. *Analytical Chemistry* **85**, 1991–1994 (2013).

[99] Prabhu, A., Venot, A. & Boons, G.-J. New Set of Orthogonal Protecting Groups for the Modular Synthesis of Heparan Sulfate Fragments. *Organic Letters* **5**, 4975–4978 (2003).

[100] Huang, Y. *et al.* Discovery of a Heparan Sulfate 3-*O*-Sulfation Specific Peeling Reaction. *Analytical Chemistry* **87**, 592–600 (2015).

[101] Vizcaíno, J. A. *et al.* 2016 update of the PRIDE database and its related tools. *Nucleic Acids Research* **44**, D447–456 (2016).

[102] Maxwell, E. *et al.* GlycReSoft: A Software Package for Automated Recognition of Glycans from LC/MS Data. *PLOS One* **7**, e45474 (2012).

[103] Wolff, J. J., Laremore, T. N., Busch, A. M., Linhardt, R. J. & Amster, I. J. Influence of Charge State and Sodium Cationization on the Electron Detachment Dissociation and Infrared Multiphoton Dissociation of Glycosaminoglycan Oligosaccharides. *Journal of the American Society for Mass Spectrometry* **19**, 790–798 (2008).

[104] Liu, J. & Pedersen, L. C. Anticoagulant Heparan Sulfate: Structural Specificity and Biosynthesis. *Applied Microbiology and Biotechnology* **74**, 263–272 (2007).

[105] Rapraeger, A. C., Guimond, S., Krufka, A. & Olwin, B. B. Regulation by heparan sulfate in fibroblast growth factor signaling. *Methods in Enzymology* **245**, 219–240 (1994).

[106] Fuster, M. M. & Wang, L. Endothelial heparan sulfate in angiogenesis. *Progress in Molecular Biology and Translational Science* **93**, 179–212 (2010).

[107] Cool, S. M. & Nurcombe, V. Heparan sulfate regulation of progenitor cell fate. *Journal of Cellular Biochemistry* **99**, 1040–1051 (2006).

[108] Yamaguchi, Y. Lecticans: Organizers of the brain extracellular matrix. *Cellular and Molecular Life Sciences* **57**, 276–289 (2000).

[109] Lemons, M. L., Howland, D. R. & Anderson, D. K. Chondroitin Sulfate Proteoglycan Immunoreactivity Increases Following Spinal Cord Injury and Transplantation. *Experimental Neurology* **160**, 51–65 (1999).

[110] Purushothaman, A., Sugahara, K. & Faissner, A. Chondroitin Sulfate "Wobble Motifs" Modulate Maintenance and Differentiation of Neural Stem Cells and Their Progeny. *The Journal of Biological Chemistry* **287**, 2935–2942 (2012).

[111] Plaas, A. H. K., West, L. A., Wong-Palms, S. & Nelson, F. R. T. Glycosaminoglycan Sulfation in Human Osteoarthritis Disease-Related Alterations at the Non-Reducing Termini of Chondroitin and Dermatan Sulfate. *Journal of Biological Chemistry* **273**, 12642–12649 (1998).

[112] Quantock, A. J., Young, R. D. & Akama, T. O. Structural and biochemical aspects of keratan sulphate in the cornea. *Cellular and Molecular Life Sciences* **67**, 891–906 (2010).

[113] Hayashi, Y. *et al.* Lumican is required for neutrophil extravasation following corneal injury and wound healing. *Journal of Cell Science* **123**, 2987–2995 (2010).

[114] Zeltz, C. *et al.* Lumican inhibits cell migration through $a2\beta1$ integrin. *Experimental Cell Research* **316**, 2922–2931 (2010).

[115] Budnik, B. A., Haselmann, K. F. & Zubarev, R. A. Electron detachment dissociation of peptide di-anions: An electron–hole recombination phenomenon. *Chemical Physics Letters* **342**, 299–302 (2001).

[116] Coon, J. J., Shabanowitz, J., Hunt, D. F. & Syka, J. E. P. Electron transfer dissociation of peptide anions. *Journal of the American Society for Mass Spectrometry* **16**, 880–882 (2005).

[117] Wolff, J. J., Laremore, T. N., Busch, A. M., Linhardt, R. J. & Amster, I. J. Electron detachment dissociation of dermatan sulfate oligosaccharides. *Journal of the American Society for Mass Spectrometry* **19**, 294–304 (2008).

[118] Wolff, J. J., Laremore, T. N., Aslam, H., Linhardt, R. J. & Amster, I. J. Electron induced dissociation of glycosaminoglycan tetrasaccharides. *Journal of the American Society for Mass Spectrometry* **19**, 1449–1458 (2008).

[119] Ruhnau, B. Eigenvector-centrality — a node-centrality? *Social Networks* **22**, 357–365 (2000).

[120] Hahn, M. W. & Kern, A. D. Comparative Genomics of Centrality and Essentiality in Three Eukaryotic Protein-Interaction Networks. *Molecular Biology and Evolution* **22**, 803–806 (2005).

[121] Park, H. W. & Thelwall, M. Hyperlink Analyses of the World Wide Web: A Review. *Journal of Computer-Mediated Communication* **8**, JCMC843 (2003).

[122] Brin, S. & Page, L. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems* **30**, 107–117 (1998).

[123] He, X., Gao, M., Kan, M.-Y. & Wang, D. BiRank: Towards Ranking on Bipartite Graphs. *IEEE Transactions on Knowledge and Data Engineering* **PP**, 1–1 (2016).

[124] Shi, X., Huang, Y., Mao, Y., Naimy, H. & Zaia, J. Tandem Mass Spectrometry of Heparan Sulfate Negative Ions: Sulfate Loss Patterns and Chemical Modification Methods for Improvement of Product Ion Profiles. *Journal of the American Society for Mass Spectrometry* **23**, 1498–1511 (2012).

[125] Hagberg, A. A., Schult, D. A. & Swart, P. J. Exploring Network Structure, Dynamics, and Function Using NetworkX. *Proceedings of the 7th Python in Science Conference (SciPy 2008)* 11–15 (2008).

[126] Arungundram, S. *et al.* Modular Synthesis of Heparan Sulfate Oligosaccharides for Structure-Activity Relationship Studies. *Journal of the American Chemical Society* **131**, 17394–17405 (2009).

[127] Zong, C. *et al.* Integrated Approach to Identify Heparan Sulfate Ligand Requirements of Robo1. *Journal of the American Chemical Society* **138**, 13059–13067 (2016).

[128] Zong, C. *et al.* Heparan Sulfate Microarray Reveals That Heparan Sulfate-Protein Binding Exhibits Different Ligand Requirements. *Journal of the American Chemical Society* **139**, 9534–9543 (2017).

[129] Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. Basic local alignment search tool. *Journal of Molecular Biology* **215**, 403–410 (1990).

[130] Subramanian, A. *et al.* Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences* **102**, 15545–15550 (2005).

[131] Raghunathan, R. *et al.* Glycomic and Proteomic changes in aging brain nigrostriatal pathway. *Molecular & Cellular Proteomics* **17**, 1778–1787 (2018).

[132] R Core Team. *R: A Language and Environment for Statistical Computing* (R Foundation for Statistical Computing, Vienna, Austria, 2015).

[133] Chang, W., Cheng, J., Allaire, J. J., Xie, Y. & McPherson, J. *Shiny: Web Application Framework for R* (2018).

[134] Hastie, T., Tibshirani, R., Narasimhan, B. & Chu, G. *Impute: Impute: Imputation for Microarray Data* (2016).

[135] Warnes, G. R. *et al. Gplots: Various R Programming Tools for Plotting Data* (2016).

[136] Tang, Y., Horikoshi, M. & Li, W. Ggfortify: Unified Interface to Visualize Statistical Result of Popular R Packages. *The R Journal* **8** (2016).

[137] Horikoshi, M. & Tang, Y. *Ggfortify: Data Visualization Tools for Statistical Analysis Results* (2018).

CURRICULUM VITAE