HAL
open science

Topics in mass spectrometry based structure determination

Deepesh Agarwal

To cite this version:

Deepesh Agarwal. Topics in mass spectrometry based structure determination. Other [cs.OH]. Université Nice Sophia Antipolis, 2015. English. NNT : 2015NICE4048 . tel-01176554

HAL Id: tel-01176554

https://theses.hal.science/tel-01176554v1

Submitted on 15 Jul 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITE NICE SOPHIA ANTIPOLIS

ECOLE DOCTORALE STIC

SCIENCES ET TECHNOLOGIES DE L"INFORMATION ET DE LA COMMUNICATION

T H E S E

pour obtenir le titre de

Docteur en Sciences

de l'Université Nice Sophia Antipolis

Mention Informatique

présentée et soutenue par

Deepesh Agarwal

Topics in Mass Spectrometry Based Structure Determination

Thèse dirigée par Frédéric CAZALS

soutenue le 18/05/2015

Jury:

Perdita Barran	Professeur, Univ. de Manchester	Rapporteur
Rumen Andonov	Professeur, Univ. de Rennes	Rapporteur
Julia Chamot-Rooke	Directeur de recherche, Institut Pasteur	Examinateur
Gilles Bernot	Professeur, Univ. de Nice Sophia Antipolis	Examinateur
Frédéric Cazals	Directeur de Recherche, Inria	Advisor

Acknowledgement

I would like to exuberantly thank my thesis supervisor, Frédéric Cazals for advising my thesis project. He always found time for discussion to ensure the progress on the project no matter how busy he was with his other commitments. He inculcated quite useful habbit of documenting the papers read, formalization of the problem, findings simultaneously. As a result of this, the papers, reports and main chapters of my thesis were written along the way. The domain of algorithm development and programming in C++, Python was new for me given that I did my undergraduate studies in Biochemical Engineering and Biotechnology. Therefore, I also thank him for his patience while I was making myself familiar with such new things.

I would like to sincerely thank David, Christelle, Julio and Stéphane from the project team COATI with whom we had a very productive collaboration on the Connectivity inference problem. They helped us in developing algorithms and obtaining theoretical proofs for the same. They also helped us in reviewing the papers, one of which finally got published in the proceeding of European symposium of algorithms (ESA) and another in the journal of Molecular and Cellular Proteomics.

I would like to express my special thanks to my thesis reviewers, Prof. Perdita Barran and Prof. Rumen Andonov who took time to read my thesis and had given crucial suggestions to be incorporated in it. It was very kind of Prof. Barran to have given detailed suggestions to fine tune the Introduction chapter on Mass Spectrometry. I appreciate the comments of my examiner, Dr. Julia Chamot-Rooke during my thesis defense to give a short illustrative overview in the beginning of the thesis.

My colleagues Tom, Alix, Noel, Andrea and then later Simon and Dorian in my team, ABS were generously helpful during my thesis. Every time I approached them for any help such as related to implementation in C++, Python an system related issues, they made sure that my problem be resolved. With Romain I have had luncheons and street cocktails and had discussed on variety of issues related to international politics, philosophy of life, sociology which were productive escape from PhD thesis at times. I am also glad to have practiced my french with them at work. It was a unique experience to be able to communicate in a new language with them. This instilled confidence in me to get things done elsewhere such as at office of administration at Inria, bank, social security.

I would also like to thank Manish and Vikash who were kind enough to cook for me at times when I was busy with work or was not in a mood to do any cooking. My stay was made pleasant by many of my colleagues from the project teams – Titane, Geometrica, Athena, Neuromathcomp, Asclepios. With them I had done several outings in and out of Nice.

Finally, I take this opportunity to express my gratitude to my parents and other family members who supported me throughout in this endeavor. They stood firmly especially at times when progress in the thesis was slow and things were seemingly difficult. It was really encouraging to hear from my parents that I made them proud and they are proud of me.

Deepesh

Contents

1 Detailed thesis overview **11**
1.1 Mass Spectrometry of Protein assemblies . . . 11
1.1.1 Problems in Mass spectrometry of protein complexes . . . 12
1.1.2 Data required from the MS for the two problems discussed . . . 12
1.2 Stoichiometry Determination (SD) problem . . . 13
1.2.1 Our contributions . . . 14
1.3 Minimum Connectivity Inference (MCI) problem . . . 15
1.4 Minimum Weight Connectivity Inference (MWCI) problem . . . 15
1.4.1 Bootstrapping procedure . . . 16

2 Resumé détaillé de la thèse **19**
2.1 La spectrométrie de masse des assemblages de protéines . . . 19
2.1.1 Problèmes en spectrométrie de masse pour les gros complexes protéiques . . . 20
2.1.2 Les données de spectrométrie de masse nécessaires aux deux problèmes discutés . . . 20
2.2 Le problème de la détermination de stoechiométrie (SD) . . . 21
2.2.1 Nos contributions . . . 22
2.3 Le problème d'inférence de connectivité minimum (MCI) . . . 23
2.4 Le problème d'inférence de la connectivité de poids minimum (MWCI) . . . 23
2.4.1 La procédure de bootstraping . . . 24

3 Introduction **27**
3.1 Problems in structural biology of large protein assemblies . . . 27
3.2 Mass Spectrometry: Overview . . . 29
3.2.1 Motivation . . . 29
3.2.2 Principles . . . 30
3.2.3 Instruments . . . 32
3.2.4 Native Mass Spectrometry . . . 37
3.2.5 Tandem Mass Spectrometry (MS/MS) . . . 38
3.2.6 Ion Mobility Mass Spectrometry . . . 39
3.3 Molecular systems handled: from chemistry to structural biology . . . 41
3.4 Data Processing . . . 42
3.4.1 Obtaining Mass Spectra . . . 42
3.4.2 Specific pipeline for Metabolites . . . 42
3.4.3 Specific pipeline for Proteins: Bottom-up approach . . . 43
3.4.4 Specific pipeline for Assemblies: Top-down approach . . . 46
3.5 Two Specific Algorithmic Challenges: Mass Decompositions and Connectivity Inference . . . 47
3.5.1 Mass Decompositions . . . 47
3.5.2 Connectivity Inference . . . 48
3.6 Thesis overview . . . 48
3.6.1 Stoichiometry determination . . . 48

3.6.2 Connectivity Inference: the Un-weighted Case . . . 49
3.6.3 Connectivity Inference: the Weighted Case . . . 49

4 Stoichiometry Determination Problems **51**
4.1 Introduction . . . 51
4.1.1 Structural Proteomics and Mass Decompositions . . . 51
4.1.2 Mass Decompositions: Float Type and Integer Type Problems . . . 52
4.1.3 Contributions . . . 53
4.2 Theory and Algorithms . . . 55
4.2.1 Denumerants, Unbounded Knapsack and Subset-sum Problems . . . 55
4.2.2 UKP and SSP: on the Number of Solutions . . . 56
4.2.3 Output Sensitive Algorithm . . . 57
4.3 Solving Float Type Problems via Tree Like Enumeration . . . 57
4.4 Solving Integer Type Problems with Dynamic Programming . . . 60
4.5 Material and Methods . . . 61
4.5.1 Comparison: Methodology . . . 61
4.5.2 Datasets for Integer Type Problems . . . 61
4.5.3 Dataset for Float Type Problems . . . 62
4.6 Results: Integer Type Problems . . . 63
4.6.1 Biological Examples: Enumeration Matters Even at Null Noise Level . . . 63
4.6.2 Counting Solutions and Convergence to the Denumerant . . . 64
4.6.3 Solution Sketches to Represent a Solution Set . . . 64
4.7 Results: Algorithm `DIOPHANTINE`– versus `DECOMP` . . . 68
4.7.1 Float Type Problems . . . 68
4.7.2 Integer Type Problems . . . 69
4.8 Results: Algorithm `DP++`– versus `DECOMP` . . . 72
4.8.1 Float Type Problems . . . 72
4.8.2 Integer Type Problems . . . 74
4.9 Results: Algorithm `DIOPHANTINE`versus `DP++` . . . 77
4.9.1 Float Type Problems . . . 77
4.9.2 Integer Type Problems . . . 77
4.10 Conclusion and Outlook . . . 78
4.11 Supplemental: Implementation . . . 80
4.11.1 Implementation sketch . . . 80
4.12 Supplemental: Material . . . 80
4.12.1 Detailed Description of Biological Complexes . . . 80
4.13 Supplemental: Algorithm `DIOPHANTINE` . . . 82
4.13.1 Output Sensitive Behavior for Integer Type Problems . . . 82
4.14 Supplemental: Algorithm `DP++` . . . 85
4.14.1 Float Type Problems . . . 85
4.14.2 Integer Type Problems . . . 87
4.14.3 Plots Corresponding to the Table 4.11 . . . 89
4.15 Supplemental: Algorithms `DP++`vs. `DECOMP` . . . 91
4.15.1 Integer Type Problems . . . 91
4.15.2 Plots Corresponding to the Table 4.12 . . . 94
4.15.3 Studying the hierarchical tree for Biological complexes . . . 98
4.15.4 Scatter plots to compare `DP++` and `DECOMP` . . . 100

5 Connectivity Inference: the Unweighted Case **101**
5.1 Introduction . . . 101
5.1.1 Connectivity Inference for Macro-molecular Assemblies . . . 101
5.1.2 Outline . . . 103
5.2 Preliminaries and Hardness . . . 104
5.2.1 Simplifying an Instance of MCI: Reduction Rules . . . 104
5.2.2 Hardness . . . 104
5.3 Solving the Problem to Optimality using Mixed Integer Linear Programming . . . 105
5.4 Approximate Solution based on a Greedy Algorithm . . . 107
5.4.1 Design and Properties . . . 107
5.4.2 Implementation . . . 108
5.5 Experimental Results . . . 109
5.5.1 Assemblies of Interest and Reference Contacts . . . 109
5.5.2 Results . . . 111
5.6 Conclusion and Outlook . . . 115
5.7 Supplemental: Statistics per Assembly . . . 116
5.7.1 Yeast Exosome . . . 116
5.7.2 Yeast Proteasome Lid . . . 117
5.7.3 eIF3 . . . 118

6 Connectivity Inference: the Weighted Case **123**
6.1 Connectivity Inference from Sets of Oligomers . . . 123
6.2 Minimum Weight Connectivity Inference: Mathematical Model . . . 125
6.3 Minimum Weight Connectivity Inference: Algorithms . . . 126
6.3.1 Algorithm `MILP-W` . . . 126
6.3.2 Solutions and consensus solutions . . . 126
6.3.3 Algorithm $\texttt{MILP-W}_{\mathrm{B}}$ . . . 127
6.4 Material: Test Systems . . . 127
6.4.1 Yeast exosome . . . 127
6.4.2 Yeast 19S Proteasome lid . . . 128
6.4.3 Human eIF3 . . . 128
6.5 Results . . . 129
6.5.1 Algorithm $\texttt{MILP-W}_{\mathrm{B}}$ . . . 129
6.5.2 Algorithm `MILP-W` . . . 130
6.6 Discussion and Outlook . . . 130
6.7 Artwork . . . 131
6.7.1 Methods . . . 131
6.7.2 Yeast Exosome . . . 133
6.7.3 Yeast Proteasome lid . . . 136
6.7.4 eIF3 . . . 138
6.8 Supplemental: Results . . . 140
6.8.1 Yeast Exosome . . . 140
6.8.2 Yeast 19S Proteasome lid . . . 142
6.8.3 Eukaryotic Translation factor eIF3 . . . 143
6.8.4 Using Weights: an Illustration . . . 145
6.9 Supplemental: Algorithms and Programs . . . 148
6.9.1 Problem hardness, existing algorithms and contributions . . . 148
6.9.2 Algorithm $\texttt{MILP-W}_{\mathrm{B}}$: pseudo-code . . . 148
6.9.3 Implementation . . . 148
6.10 Supplemental: Using Weights: a Detailed Study . . . 149
6.10.1 Methods . . . 149
6.10.2 Results . . . 150

7 Conclusion **159**

8 Supplemental: Biological Systems Studied **169**
8.1 Supplemental: Lists of Oligomers for the Assemblies Studied . . . 169
8.1.1 Yeast Exosome . . . 169
8.1.2 Yeast 19S Proteasome lid . . . 170
8.1.3 Eukaryotic Translation factor eIF3 . . . 170
8.2 Supplemental: Reference Contacts Within Assemblies . . . 171
8.2.1 Pairwise Contacts within Macro-molecular Oligomers . . . 171
8.2.2 Yeast Exosome . . . 173
8.2.3 Yeast Proteasome Lid . . . 174
8.2.4 eIF3 . . . 175

Glossary

Oligomers (or sub-complexes) are small assemblies dissociated from the intact assembly.

Error or Noise in the mass measurement is the shift in the mass due to solvent or buffer molecules attached to the protein complexes.

Corollary: 0% noise in the mass measurement would correspond to the average mass considering the abundance of the isotopes of the constituent elements. Also, it corresponds to the measurement of mass with complete desolvation.

Nodes in the graph corresponds to the protein subunits

Edges in the graph corresponds to the contacts between protein subunits in the assembly.

Subgraph corresponds to an oligomer (or a sub-complex).

Chapter 1

Detailed thesis overview

1.1 Mass Spectrometry of Protein assemblies

Mass spectrometry is an old classic technique to measure masses of the molecules such as metabolites, proteins and peptides. The principle steps which molecules undergo for their mass determination are:
Ionization: The molecules are introduced into the mass spectrometer as ions. For the biological molecules, usually soft ionization techniques such as Electrospray ionization and MALDI are employed to prevent high degree of fragmentation.
Ion segregation: The ions are separated by the mass analyzer on the basis of their mass-to-charge (m/z) ratio.
Ion detection: The ions are electrically detected by the detector and m/z spectrum is recorded.

The classical method to study biological molecules is bottom-up mass spectrometry, which analyze the peptides generated by enzymatic or chemical digestion of proteins. In contrast, the top-down mass spectrometry is a recent advancement which aims at study of the intact assemblies. Protein assemblies (or complexes) are ensemble of protein subunits held together by non-covalent interactions. Their masses can range from several hundred kDa (kilodaltons) to MDa (megadaltons). These protein assemblies were called "Molecular Elephants" by John Fenn (Fig. 1.1). The contributions to the electrospray ionization of protein complexes ([FMM$^+$89]) have made John Fenn won the Nobel prize in chemistry along with Koichi Tanaka in 2002.

Figure 1.1: **Molecular Elephants.** John Fenn called protein assemblies as molecular elephants in his Nobel Lecture: *Electrospray wings for molecular elephants.*

It is to be noted that the mass measured through mass spectrometer is accompanied with some error in the measurement which could arise due to water and/or buffer adducts attached to the molecules under scrutiny. Other factor contributing to the uncertainty in the mass measurement is the post-translational modifications of the individual proteins. Concerning the precision of the instrument, with improvement in the desolvation, orbitrap analyzer has shown to measure mass with the accuracy of 50ppm, although this

is done on few assembly of proteins. More conventionally, quadrapole coupled to Time-of-flight (Q-TOF) analyzer is used for which the mass shift could be upto 1% for the assemblies upto 1 MDa.

1.1.1 Problems in Mass spectrometry of protein complexes

In the context of Top-down Mass Spectrometry of protein complexes, various problems arise concerning the study of architecture of protein assemblies:

- *Stoichiometry Determination (SD):* In a heteromeric protein complex, the problem is to determine the copy number of each protein type in the assembly. The SD problem is called an *interval SD problem* when the intact mass of the assembly is not exactly known due to the chemical noise, that is belongs to an interval.

- *Connectivity Inference (CI):* As discussed above, a protein assembly is held together by non-covalent interactions. The problem of connectivity inference is to find all such protein-protein pairwise interactions within the assembly.

- *Macromolecular Packing and Shape:* Ion-mobility mass spectrometry can be used to provide with the full description of the shape.

- *Assembly Dynamics:* During the assembly process of a protein assembly, MS can be used to capture the population of intermediate states. The decrease in the intensity of monomers and the increase in the intensity of the intact assembly provides information on timescales and kinetics of the assembly process.

In this thesis, we focus on the two problems namely, the Stoichiometry determination problem and the Connectivity Inference problem.

1.1.2 Data required from the MS for the two problems discussed

The protein assemblies under scrutiny are treated in different solution conditions to get input for the two problems of interest above. To measure the mass of the intact assembly, native MS is performed under non-denaturing conditions with buffered aqueous solution mimicking the physiological conditions while being compatible with electrospray ionization requirements.

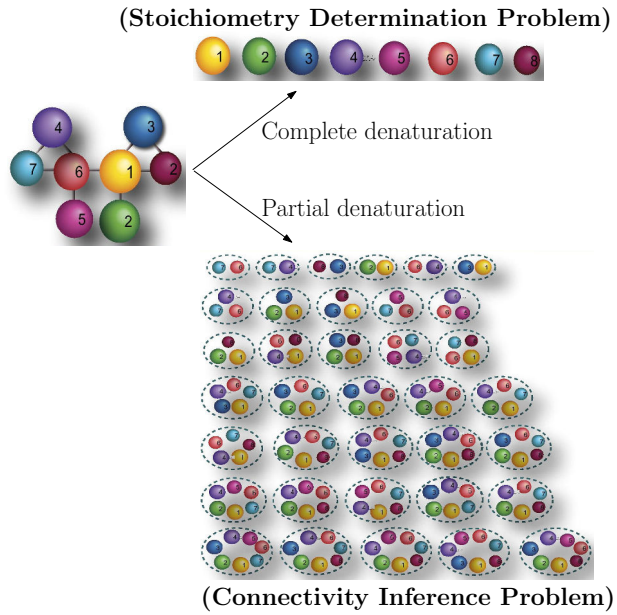


Figure 1.2: **The generation of the data for the two problems discussed in this thesis is done by complete or partial denaturation of the assembly.**

For a heteromeric protein assembly, to get the exact masses of the individual subunits, a denatured solution of complex is undergone the MS analysis. As a result, all the non-covalent bonds are disrupted yielding individual subunits. This provides the input for the Stoichiometry determination (SD) problem which is to determine the copy number of each protein type such that their added mass equals the mass of the intact assembly (Fig. 1.2).

To generate the necessary data for the connectivity inference, the complex-containing solution is treated under soft denaturing conditions. In such conditions, the partial denaturation of the assembly occurs yielding a population of overlapping sub-complexes (Fig. 1.2). The composition for each sub-complex is determined by solving the SD problem. The Connectivity Inference (CI) problem then is to infer the connectivity of proteins within the assembly complying with the overlapping sub-complexes generated.

1.2 Stoichiometry Determination (SD) problem

Considering a heteromeric protein complex of p type of proteins. Let the mass of the intact assembly measured is M. The weight vector, W enclosing the masses for p type of proteins is:

$$W = \{w_1, w_2, w_3, ..., w_p\}$$

Taking into account the error in the measurement and defining s_i, the stoichiometry of the ith type of protein (i.e., the number of copies of this protein), the interval SD problem is defined by the following equation:

$$| \sum_{i=1,2,..,p} s_i w_i - M | \quad \leq \quad \varepsilon, \text{ where } \varepsilon \text{ is the chemical noise} \tag{1.1}$$

The above problem defines the SD problem for an interval. The exact problem is a particular case of the interval SD problem, when $\varepsilon = 0$. The existing algorithms to solve the exact problem are `DECOMP` and another algorithm based on dynamic programming [BL05a]. Both these algorithm require some pre-processing to construct the table. It is then followed by the enumeration of solutions through an enumeration-tree using

the information from the table. The time complexity of the enumeration step is dependent on the number of solutions, so these are output sensitive algorithms.

Previously, the interval problem has been tackled by solving all exact problems in the interval $[M - \varepsilon, M + \varepsilon]$. The downside of this approach is that there are redundant computations because all the exact problems for the interval are solved independently and the union of all the stoichiometry vectors are finally reported.

1.2.1 Our contributions

We propose a constant-space enumeration algorithm `DIOPHANTINE` and another algorithm `DP++` which is based on dynamic programming. Both these algorithms, enumerate solutions using a single enumeration tree unlike previous algorithms which generate an enumeration tree for each exact problem in the interval. We illustrate the enumeration by `DIOPHANTINE` using a simple example in the Fig. 1.3.

Suppose we want to decompose $M = 20$ Da into the weight vector, $W = \{11, 7, 5\}$ Da with chemical noise of 5%, i.e., $\varepsilon = 1$ Da. The problem can be rephrased as follows: Decompose $M = 21$ Da into the weight vector, $W = \{11, 7, 5\}$ Da within the remainder of 2 Da (2ε).

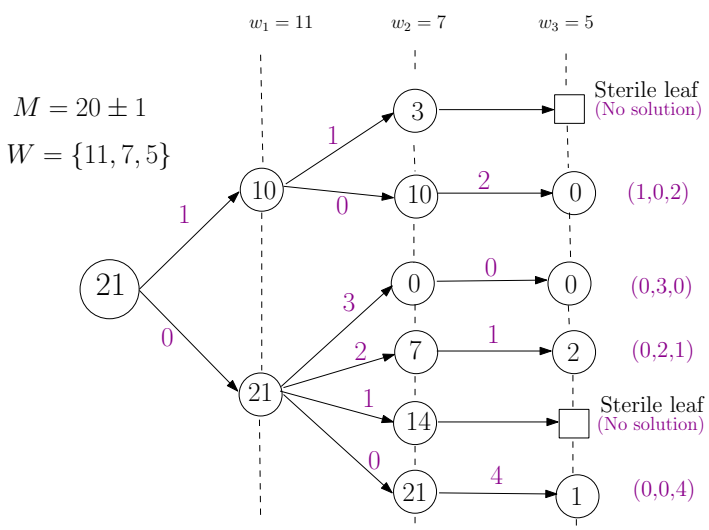


Figure 1.3: **Illustration of an enumeration tree used by the algorithm `DIOPHANTINE` for an interval problem.** Decompose 20 into {11, 7, 5} with $\varepsilon = 1$. Note that sterile leaves are formed when there is no feasible solution to the problem.

Enumeration by `DIOPHANTINE` The enumeration tree is shown in the Fig. 1.3. All the nodes in the tree correspond to the remaining mass after the decomposition carried out up to that level in the tree, successively using masses of 11, 7 and 5 Da. A solution is a vector of stoichiometries; the entries in this vector are the copy numbers of the individual proteins, each such value being associated with an edge of the tree. A sterile leaf is formed if there is no feasible decomposition found.

The algorithm `DP++` requires to form a binary table prior to the enumeration. It stores information whether a path in the tree leads to solution or a sterile leaf. Following this additional information, in the enumeration tree, only the paths leading to the solutions are formed avoiding futile computations. That is why, `DP++` is output sensitive algorithm since the number of computations are proportional to the number of solutions.

Practically, our algorithms are faster than the previous algorithms by a factor up to 3-4 order of magnitude.

1.3 Minimum Connectivity Inference (MCI) problem

The second problem that we worked upon is the Connectivity inference (CI) problem. Consider a population of overlapping sub-complexes generated after the treatment of complex containing solution with a soft denaturant. Since, we do not know the number of protein-protein contacts in the assembly apriori, we solve MCI to find the minimum number of contacts that comply with the list of oligomers.

Briefly, each protein subunit is considered as a node in the graph and edges (contacts) are sought in the graph of all the protein subunits.

The MCI problem consists of finding a set of edges of minimal cardinality such as each subgraph corresponding to an oligomer is connected.

Usually, there are multiple solutions to the MCI. Since each solution is optimal and is equally likely, therefore final set of contacts is the union of all contacts in the set of solutions. It is illustrated using a simple lego example in the Fig. 1.4.

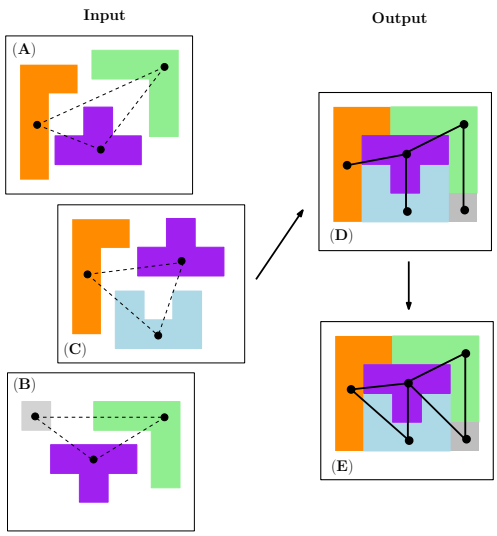


Figure 1.4: **Illustration of the Connectivity Inference problem.** (A)-(B) Three oligomers composed from 5 type of legos (proteins); (D) One possible optimal solution of MCI; (E) Set of edges (pairwise interactions) upon combining all optimal solutions.

To solve the MCI problem, we have proposed a Mixed Integer Linear Programming (`MILP`) formulation which provides the exact solution and a `Greedy` algorithm which is sub-optimal but is faster than `MILP`. The algorithm `MILP` uses an objective function which is the sum of binary variables over all the possible contacts in the graph described above. The algorithm `Greedy` iteratively chooses the contacts in a greedy manner, based on their abilities to connect pairs of proteins in different oligomers.

1.4 Minimum Weight Connectivity Inference (MWCI) problem

While solving MCI, we assumed that all the contacts are equally likely and no contact has any apriori bias over the other. However, some apriori information on the contacts in the assembly can be obtained using biochemical and biophysical experiments such as yeast-two-hybrid essays and immuno co-precipitation essays. Thus, the likelihood of contacts derived from the experiments could be used to infer the connectivity. We assign weight to contacts in the interval $[0, 1]$. The default value is chosen to be 0.5. Higher weight means higher likelihood and vice-versa. We therefore, define a Minimum weight connectivity inference (MWCI) problem. The objective function is modified to have two components: the first is the sum of binary variables over the edges (contacts); the second uses weight of the contacts. The relative predominance of the

components is regulated by a fractional factor α. Note that MCI is a special case of MWCI when $\alpha = 1$. The exact solutions of the MWCI problem are found by the algorithm `MILP-W`. Firstly, an optimal solution is found, then other solutions are found by including a constraint to prevent the sampling of previously found solutions.

As discussed above, solving MWCI (or MCI) usually yields a set of solutions. A solution is a set of contacts (edges in the graph). We define some scores to analyze the set of solutions. The score of a contact is the count of solutions it is a part of. After finding the scores of all the contacts, the score of a solution is the sum of the scores of its contacts. Among these solutions, the highest scoring solutions are called consensus solutions. The consensus contacts are the union of contacts in the set of consensus solutions. When the quality of solutions are examined on a test system for which experimentally determined contacts are at disposal, e.g., yeast exosome, it is observed that there are very few false positives in the set of solutions. Further, the consensus solutions have less number of contacts but even fewer false positives.

1.4.1 Bootstrapping procedure

The consensus contacts are the backbone of the set of the solutions and also they are of high specificity. In order to find more such contacts, i.e., to go beyond the consensus contacts associated with consensus solutions, we propose a bootstrapping procedure, $\texttt{MILP-W}_{\mathrm{B}}$. It is illustrated using a simple example shown in the Fig. 1.5. Suppose there are six oligomers composed from 6 nodes. The algorithm `MILP-W` is run with $\alpha = 1$, essentially solving MCI in this case. (Other value of α can be chosen.) Upon taking the union of all contacts in the solution set, it yields 10 contacts among which 6 are consensus contacts shown as bold in the Fig. 1.5. Then the consensus contacts are precluded one by one from the pool of contacts to be sampled and same problem of 6 oligomers is solved using `MILP-W` with same α used initially. On precluding the contact $\{1, 3\}$, the solution set now has two more consensus contacts, $\{1, 5\}$ and $\{2, 3\}$. Similarly, other consensus contacts from the initial set are precluded and new consensus contacts are found. Note that, more than one consensus contact can also be precluded at a time. However, this increases the chance of sampling of false positives–the contacts compatible with the oligomers but do not exist within the assembly.

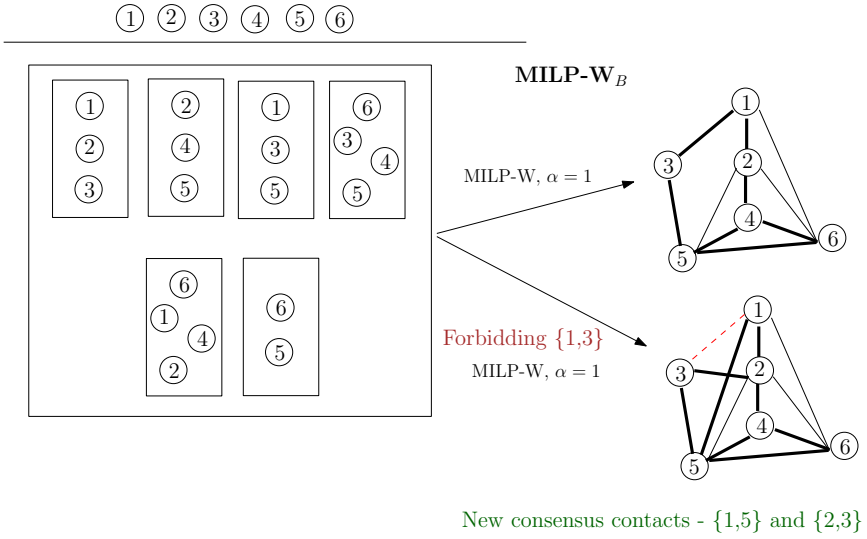


Figure 1.5: **Bootstrap illustration on a simple example**

We have run our algorithms on mass spectrometry data of three assemblies (*yeast exosome, yeast 19S proteasome lid and eIF3*). The experimentally validated contacts are available for these systems either coming from various experiments (x-ray crystal structure, cryo-EM, cross-linking). Our solutions have been

compared with those generated by an algorithm "Network Inference" (`NI`), previously proposed to solve the MCI problem [THS+08]. Upon comparing the quality of solutions with this algorithm, our results show increase in the sensitivity of about 1.7 times without trading off with the specificity.

The data acquisition using mass spectrometry and all the algorithms for the two problems are discussed in details in the subsequent chapters.

Chapter 2

Resumé détaillé de la thèse

2.1 La spectrométrie de masse des assemblages de protéines

La spectrométrie de masse est une technique éprouvée pour mesurer les masses de molécules comme les métabolites, les protéines ou les peptides. Les étapes principales que les molécules subissent pour la détermination de leur masse sont:

- *l'ionisation:* les molécules sont introduites dans le spectromètre de masse en tant qu'ions. Pour les molécules biologiques, des techniques d'ionisation douces telles que l'ionisation par électronébulisation (ESI) et MALDI sont utilisées, afin d'empêcher leur fragmentation.
- *la ségrégation d'ions:* les ions sont séparés par l'analyseur de masse, en fonction de leur ratio masse sur charge (m/z).
- *la détection d'ions:* les ions sont électriquement détectés par le détecteur, et le spectre de m/z est enregistré.

Pour les molécules biologiques, la modalité classique est la *spectrométrie de masse descendante*, qui analyse les peptides générés par digestion enzymatique ou chimique des protéines. Par opposition, la *spectrométrie de masse ascendante*, visant à étudier les assemblages intacts, est une avancée récente. De tels assemblages de protéines sont constitués de sous-unités de protéines maintenues ensemble par des interactions non-covalentes. Leur masse peut varier de plusieurs centaines kDa (kilodaltons) à MDa (mégadaltons).

Ces assemblages ont été appelés "éléphants moléculaires" par John Fenn (Fig. 2.1). Les contributions à l'ionisation par électronébulisation de tels complexes ([FMM+89]) ont permis a John Fenn d'être lauréat du prix Nobel de chimie, avec Koichi Tanaka, en 2002.

Figure 2.1: **Eléphants moléculaires.** Pendant l'exposé qu'il donna lors de la remise de son prix Nobel, *Electrospray wings for molecular elephants*, John Fenn a employé l'image "élépants moleculaires" pour évoquer les gros assemblages protéiques.

Il faut noter que la masse mesurée par le spectromètre de masse s'accompagne d'erreurs de mesure, qui sont le fait d'eau et/ou des produits du tampon utilisé, qui s'attachent aux molécules étudiées. Un autre facteur contribuant à l'incertitude dans la mesure de masse est l'ensemble des modifications post-traductionnelles subies par les protéines individuelles. Concernant la précision de l'instrument, avec l'amélioration de la désolvatation, une précision de l'analyseur de 0.005% (ou 50ppm) a été atteinte (dans de rares cas) avec un instrument de type orbitrap. Plus classiquement, un quadrapole couplé à un analyseur à temps-de-vol (Q-TOF) génère une erreur pouvant aller jusqu'à 1% de la masse mésurée pour des assemblages dont la masse est de l'ordre de 1 MDa.

2.1.1 Problèmes en spectrométrie de masse pour les gros complexes protéiques

Dans le contexte de la spectrométrie de masse descendante de complexes protéiques, divers problèmes se posent pour étudier l'architecture des tels assemblages:

- *La détermination de stoechiométrie (SD):* Dans un complexe protéique de type hetero-multimère (les protéines impliquées diffèrent), le problème est de déterminer le nombre de copies de chacun des types de protéines en tenant compte de l'incertitude dans le mesure de masse. Le problème SD est qualifié de *problème par intervalle* lorsque la masse cible n'est pas connue exactement, i.e. appartient à un intervalle, ce qui est généralement le cas.

- *L'inférence de connectivitée (CI):* Comme discuté ci-dessus, un assemblage de protéines est un ensemble dont la cohésion est dûe à des interactions non-covalentes. Le problème de l'inférence de connectivité est de trouver les interactions entre paires de protéines au sein de l'assemblage.

- *L'inférence de morphologie:* La spectrométrie de mobilité ionique (IM-MS) peut être utilisée pour fournir la description complète de la forme de l'assemblage.

- *La dynamique de l'assemblage:* Pendant le processus d'assemblage d'un complexe multi-protéique, la spectrométrie de masse peut être utilisée pour capturer des états intermédiaires. La diminution de l'intensité des monomères et l'augmentation de l'intensité de l'assemblage intact fournissent des informations sur les délais et la cinétique du processus d'assemblage.

Dans cette thèse, nous nous concentrons sur deux problèmes spécifiques, à savoir la détermination de stoechiométrie et l'inférence de connectivité.

2.1.2 Les données de spectrométrie de masse nécessaires aux deux problèmes discutés

Considérons un assemblage auquel on souhaite appliquer les deux techniques d'étude ci-dessus. Un tel assemblage est traité en solution, avec des conditions différentes, pour obtenir les données utilisées comme entrées pour les deux problèmes. Pour mesurer la masse de l'assemblage intact, des mesures de spectrométrie de masse native sont effectuées à partir de la solution sans dénaturant, la solution étant tamponnée afin d'imiter les conditions physiologiques, tout en étant compatible avec les exigences de l'ionisation par électronébulisation.

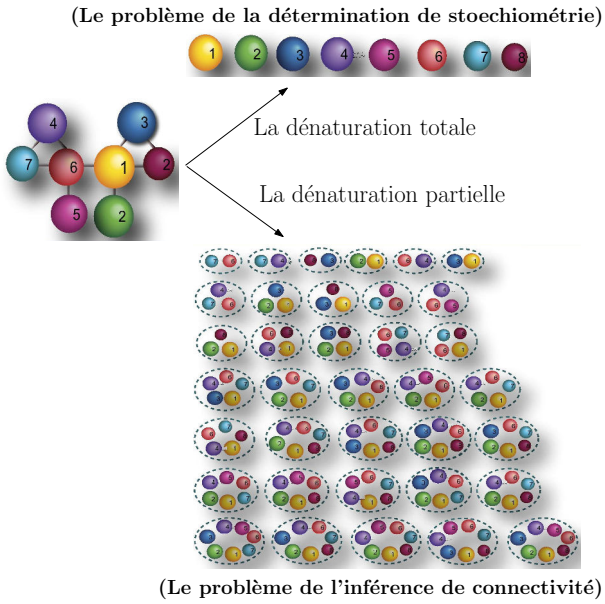


Figure 2.2: **Génération des données associées aux deux problèmes traités dans cette thèse, par dénaturation partielle ou totale d'un assemblage.**

Pour un assemblage hétéro-multimérique, pour obtenir la masse exacte de chacune des sous-unités, une analyse de masse est effectuée sur une solution dénaturée. En conséquence, tous les liaisons non-covalentes sont rompues, libérant les sous-unités individuelles. Ceci fournit l'entrée pour le problème de la détermination de stoechiométrie (SD, Fig. 2.2).

Pour générer les données nécessaire à l'inférence de connectivité, la solution contenant le complexe est traitée sous conditions dénaturantes douces. Dans de telles conditions, la dénaturation partielle de l'assemblage produit une population de sous-complexes, aussi appelés *oligomères*, en général non disjoints (Fig. 2.2). La composition de chaque sous-complexe est déterminée en résolvant le problème de SD. Le problème de l'inférence de connectivité (CI) consiste alors à inférer la connectivité des protéines au sein de l'assemblage respectant les sous-complexes générés.

2.2 Le problème de la détermination de stoechiométrie (SD)

Considérons un complexe hétéro-multimérique de protéines, impliquant des protéines de p types différents. Soit M la masse mésurée de l'assemblage intact. Le vecteur de poids, noté W, contenant les masses pour p type de protéines est noté:

$$W = \{w_1, w_2, w_3, ..., w_p\}.$$

En prenant en compte les erreurs de mesure, et en notant s_i la stoichiométrie du i-eme type de protéine (i.e. le nombre de copies de cette protéine), le problème SD par intervalle est défini par l'équation suivante:

$$| \sum_{i=1,2,..,p} s_i w_i - M | \quad \leq \quad \varepsilon; \quad \text{avec, } \varepsilon \text{ l'erreur sur la masse cible} \tag{2.1}$$

Noter que lorsque $\varepsilon = 0$, le problème par intervalle définit un problème exact. Les algorithmes existants pour résoudre le problème exact sont `DECOMP` et un autre algorithme basé sur la programmation dynamique [BL05a]. Ces deux méthodes nécessitent un pré-traitement pour construire une *table*. Elles énumèrent ensuite

les solutions grâce à un arbre d'énumération utilisant cette table. La complexité du temps de l'énumération dépendant du nombre de solutions, ces algorithmes sont dits *sensibles à la sortie*.

Avant nos travaux, les problèmes sur un intervalle étaient abordés en résolvant tous les problèmes exacts dans l'intervalle $[M-\varepsilon, M+\varepsilon]$. L'inconvénient de cette approche est qu'elle conduit à des calculs redondants car les problèmes exacts sont résolus indépendamment.

2.2.1 Nos contributions

Pour l'énumération des solutions d'un probleme SD sur un intervalle, nous proposons l'algorithme `DIOPHANTINE` nécessitant un espace mémoire constant, et l'algorithme `DP++` basé sur la programmation dynamique. Ces deux algorithmes, énumérent les solutions à l'aide d'un seul arbre d'énumération, contrairement aux algorithmes précédents qui génèrent un arbre pour chaque problème exact dans l'intervalle. Nous illustrons l'énumération par `DIOPHANTINE` à l'aide d'un exemple simple dans la Fig. 2.3.

Supposons que l'on veuille décomposer la masse $M = 20$ Da en $W = \{11, 7, 5\}$ Da avec un niveau de bruit de 5%, c.-à-d., $\varepsilon = 1$ Da. Comme nous le verrons plus tard, le problème peut être traduit comme suit: décomposer $M = 21$ Da en utilisant le vecteur de poids $W = \{11, 7, 5\}$ Da, avec un écart de 2 Da (2ε).

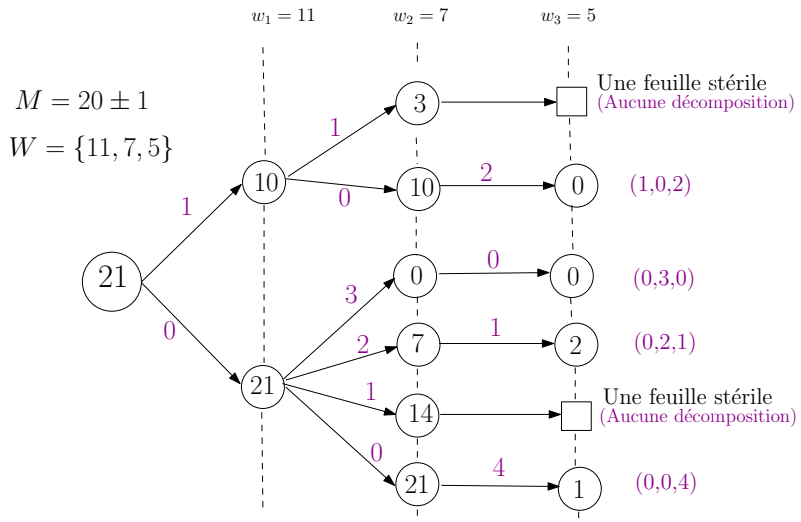


Figure 2.3: **L'illustration d'un arbre d'énumeration utilisé par l'algorithme `DIOPHANTINE` pour un problème d'intervalle.** Decomposer 20 en {11, 7, 5} avec $\varepsilon = 1$. Noter qu'une feuille stérile est formée quand il n'y a aucune solution faisable.

L'énumeration par `DIOPHANTINE`*:* L'arbre d'énumération est illustré par la Fig. 2.3. Tous les nœuds de l'arbre correspondent à la masse qui reste après la décomposition effectuée jusqu'à ce niveau. Noter que les trois niveaux utilisent successivement les masses 11, 7 et 5 Da. Une solution est un vecteur de stoechiométries; les entrées de ce vecteur correspondent aux valeurs sur les arêtes de l'arbre, entre la racine et une feuille. Une feuille stérile est formée s'il n'y a aucune décomposition faisable.

L'énumeration par `DP++`*:* L'algorithme `DP++` exige de former un tableau binaire préalablement à l'énumeration. Il stocke les informations indiquant si un chemin dans l'arbre conduit à une solution ou une feuille stérile. Grace à cette information supplémentaire, l'énumeration parvient à se concentrer sur les solutions, évitant les calculs futiles. C'est pourquoi, l'algorithme `DP++` est sensible à la sortie, le nombre d'opérations étant proportionnel au nombre de solutions.

En pratique, nos algorithmes améliorent l'état de l'art dans un rapport pouvant aller jusqu'à 3-4 ordres de grandeur.

2.3 Le problème d'inférence de connectivité minimum (MCI)

Le deuxième problème auquel nous nous sommes intéressés est l'inférence de connectivité (CI). Prenons une population de sous-complexes chevauchants générés après le traitement de la solution contenant le complexe avec un dénaturant doux. Le nombre de contacts protéine-protéine dans l'assemblage étant inconnu a priori, nous introduisons le probleme *d'inférence de la connectivité minimum*, afin de trouver le nombre minimum de contacts compatible avec la connectivité de chaque oligomère.

En bref, chaque sous-unité de protéine est considérée comme un nœud dans le graphe dont les arêtes, correspondant aux contacts, sont recherchées dans le graphe complet avec toutes les sous-unités.

Le problème MCI consiste à trouver un ensemble d'arêtes de taille minimale, de telle sorte qu'en restreignant ces arêtes aux sommets d'un oligomère, cela définisse un sous-graphe connexe.

Habituellement, il y a plusieurs solutions optimales au MCI. Toutes étant équiprobables, l'ensemble final de contacts retourné est l'union de tous les contacts dans l'ensemble des solutions. Ceci est illustré à l'aide d'un exemple simple de legos dans la Fig. 2.4.

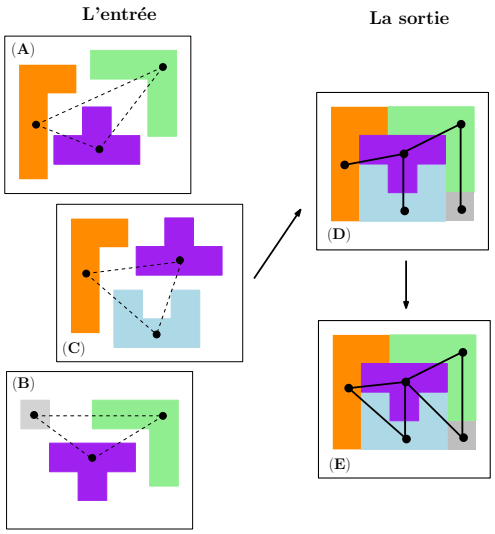


Figure 2.4: **L'illustration du problème de l'inférence de la connectivité.** (A)-(C) Les trois oligomères composés de cinq types de de legos (protéines); (D) Une solution optimale possible de MCI; (E) L'ensemnble des arêtes (contacts) en prenant l'union de toutes les solutions optimales.

Pour résoudre le problème de MCI, nous avons proposé une formulation de programmation linéaire mixte en nombres entiers (`MILP`) qui fournit les solutions exactes, et un autre algorithme (`Greedy`) qui est sous-optimal, mais polynomial. L'algorithme `MILP` utilise une fonction objectif qui est la somme des variables binaires sur tous les contacts possibles dans le graphe décrit ci-dessus. L'algorithme `Greedy` choisit itérativement les contacts de façon gloutonne, en fonction de leur aptitude à connecter des paires de protéines dans les différents oligomères.

2.4 Le problème d'inférence de la connectivité de poids minimum (MWCI)

Pour le probleme MCI, tous les contacts sont supposés équiprobables, ou autrement dit, aucun contact n'a une priorité supérieure à celle des autres.

Toutefois, certaines informations sur la plausibilité des contacts peuvent être obtenues à l'aide de diverses expériences biochimiques et biophysiques telles que la technique de double hybride ou encore la technique de

l'immuno-co-précipitation. Par conséquent, la probabilité de contacts estimée à partir de telles expériences peut être utilisée pour déduire la connectivité. On affecte ainsi un poids à chaque contact, i.e un nombre dans l'intervalle $[0, 1]$. La valeur par défaut choisie est 0,5. Une valeur proche de 1 signifie une probabilité plus élevée, et vice-versa. A partir de ces poids, nous définissons le problème d'inférence de connectivité de poids minimum (MWCI).

La fonction objectif utilisée dans MCI est modifiée pour avoir deux composantes: la première est la somme des variables binaires sur les arêtes; la seconde utilise les poids des contacts. L'importance relative de ces deux composantes est reglée grâce à un paramètre $\alpha \in [0, 1]$. Il s'avère que MCI est un cas spécial de MWCI lorsque $\alpha = 1$. Les solutions exactes du problème de MWCI sont trouvées par l'algorithme `MILP-W`. Tout d'abord une solution optimale est trouvée, puis d'autres solutions sont trouvées en incluant une contrainte pour empêcher la redécouverte de solutions trouvées précédemment.

Tel que discuté ci-dessus, la solution d'un probleme MWCI (ou MCI) est un ensemble de solutions. Une solution est un ensemble de contacts (les arêtes du graphe). Nous définissons certains scores pour analyser l'ensemble de solutions. Le score d'un contact est le nombre de solutions dont ce contact fait partie. Après avoir trouvé les scores de tous les contacts, le score d'une solution est défini comme la somme des scores de ses contacts. Ainsi, les solutions réalisant le plus haut score s'appellent les solutions consensus. Les contacts utilisés par les solutions consensus sont nommés les contacts consensus.

Lorsque la qualité des solutions est examinée sur un système test pour lequel les contacts déterminés expérimentalement sont à la disposition, par exemple *yeast exosome*, il est observé qu'il existe très peu de contacts qui soient des faux positifs. En outre, les solutions consensus utilisent moins de contacts, mais ont encore moins de faux positifs.

2.4.1 La procédure de bootstraping

Les contacts consensus peuvent être vus comme *l'épine dorsale* de l'ensemble des solutions et ils sont aussi de haute spécificité. Afin de trouver plus de tels contacts, i.e. d'aller au-delà des consensus contacts associés aux solutions consensus, nous proposons une procédure de bootstrap, $\texttt{MILP-W}_{\mathrm{B}}$. Ceci est illustré à l'aide d'un exemple simple dans la Fig. 2.5. Supposons qu'il existe six oligomères composés de 6 nœuds. L'algorithme `MILP-W` est exécuté avec $\alpha = 1$, de telle sorte que l'on résout le problème MCI dans ce cas. (D'autres valeurs de α peuvent être considerées.) En prenant l'union de tous les contacts dans l'ensemble de solutions, on a 10 contacts parmi lesquels 6 sont les contacts de consensus indiqués en gras dans la Fig. 2.5. Puis les contacts de consensus sont exclus un par un du pool de contacts à échantillonner et le même problème de 6 oligomèrs est résolu par `MILP-W`. En excluant le contact $\{1, 3\}$, l'ensemble de solutions a maintenant deux consensus contacts en plus, $\{1, 5\}$ et $\{2, 3\}$. En itérant le processus, d'autres contacts de consensus de l'ensemble initial sont exclus et de nouveaux contacts de consensus sont trouvés. Notons que plusieurs contacts de consensus peuvent être exclus en même temps. Toutefois, cela augmente la risque de trouver des faux positifs–des contacts compatibles avec les oligomères mais n'existant pas dans l'assemblage.

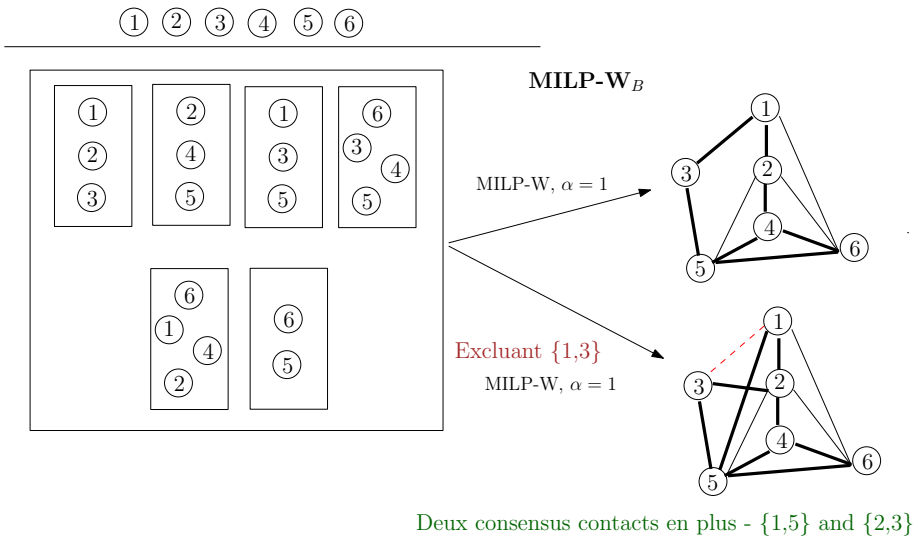


Figure 2.5: **L'illustration de la méthode de bootstrap sur un exemple simple**

Nous avons exécuté nos algorithmes sur les données de spectrométrie de masse de trois assemblages (*yeast exosome, yeast protéasome 19S lid, eIF3*). Pour ces systèmes, des contacts validés expérimentalement sont disponibles, venant de diverses expériences (cristallographie aux rayons X, cryo-EM, cross-linking). Nos solutions ont été comparées à celles générées par la métohde "Network Inference" (`NI`), préalablement proposée pour résoudre le problème de MCI [THS+08]. Par rapport aux solutions de cet algorithme, nos résultats présentent une augmentation de la sensibilité d'un facteur 1,7 environ, sans chute de la spécificité.

L'acquisition de données à l'aide de la spectrométrie de masse et tous les algorithmes pour les deux problèmes sont discutés en détail dans les chapitres suivants.

Chapter 3

Introduction

3.1 Problems in structural biology of large protein assemblies

A number of fundamental processes in the cell such as cell division (by Septalsome), protein synthesis (by Ribosome), transportation of RNA and ribosomal proteins in and out of nucleus (by Nuclear pore complex), degradation of RNA (by RNA exosome), degradation of proteins (by Proteasome), are carried out by the functional modules that are assembly of macromolecules such as proteins [RSB07]. Their size can range from several hundred kDa[1] (e.g., Yeast RNA exosome, 400 kDa) to megadaltons (e.g., 26S proteasome, 2.5 MDa). In most cases such assemblies have several different proteins and even multiple copies of the same proteins within the same assembly. One of the key endeavors in structural biology is to understand the mediation of structural machinery of such large macromolecules in their functioning. The task is challenging not only because of the complexity due to the heterogenous composition but also due to their sheer size, which makes it difficult for biochemical purification and further analysis.

There are four aspects to a biological macromolecule for its complete characterization: cellular localization, three dimensional (3D) structure, dynamics and its function (Fig. 3.1). In this context, broadly there is a two fold challenge that one faces. Firstly, the measurement of any of the property has influence on the measurement of other three, i.e. all four of them can not be precisely measured at the same time. For instance, in order to perform x-ray crystallography on the protein sample, one would require to do *in vitro* manipulation that would render the loss of information on cellular localization. On the other hand, measurement of cellular localization through *in vivo* methods although returns positional information precisely but generally yield low resolution information on the structure. Secondly, measurement of any one property comprehensively is itself elusive. For instance, dynamical nature of protein complexes can have wide range of timescales. Atomic vibrations and molecular tumbling occur at femtosecond scale, whereas, the assembling of subunits before it becomes an intact macromolecular assembly can take several minutes. No single experiment yields measurements covering such a wide range of timescales. These two aforementioned challenges require us to study the structure-function relationship of the protein assemblies through a synergy of experiments [SR11].

[1]The unified atomic mass unit, also known as Dalton, is approximately equal to the mass of one neutron or one proton.

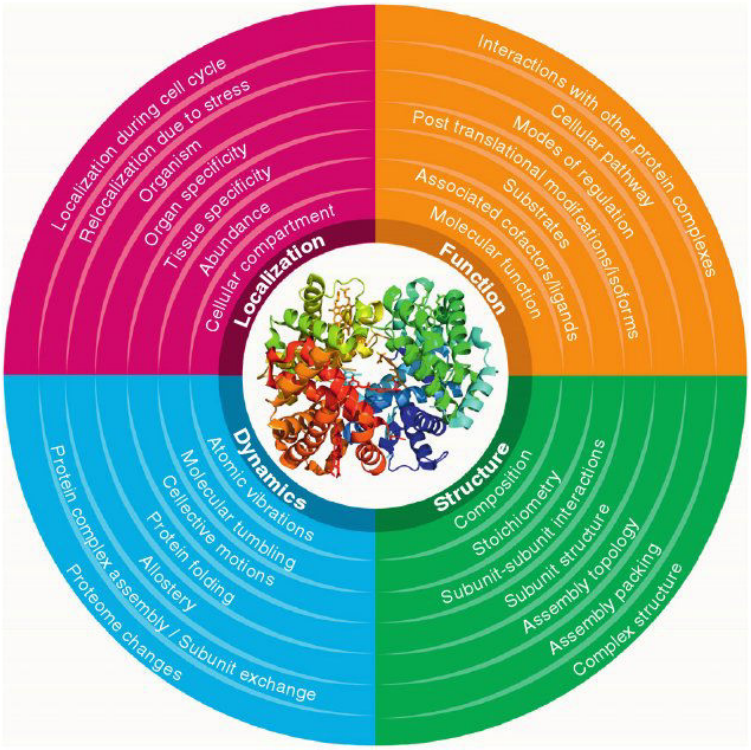


Figure 3.1: **Four attributes to completely characterize a biological macromolecule.** Picture from [SR11].

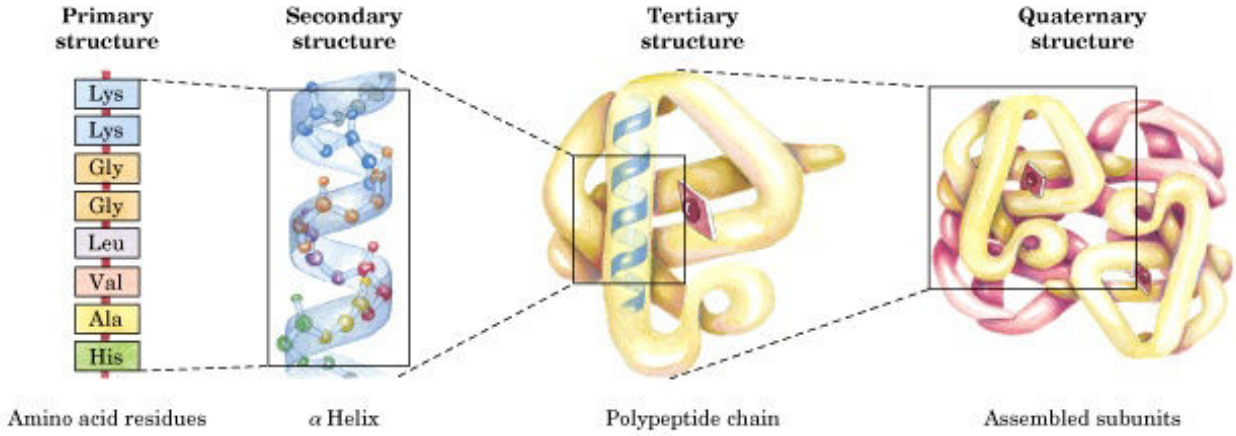


Figure 3.2: **Four levels of a protein structure.** The mass of a protein is due to its primary structure, i.e. sum of masses of composing amino acids. Relative arrangement of proteins and pairwise interactions between the proteins in the quaternary structure regulate various cellular functions. Picture from http://xray.bmc.uu.se/kurs/BSBX2/practicals/practical_2/practical_2.html.

The structure of proteins is described in four hierarchical levels of organization - primary, secondary, tertiary and quaternary structure. The primary structure is the sequence of its constituting amino acid residues in the polypeptide chain; the secondary structure is due to the localized interactions yielding various

kinds of secondary structure. Without any stablizing interactions, the polypeptide chain assumes random coil, however in the presence of hydrogen bonds between some residues of the chain, it assumes either $\alpha-$helix or planar $\beta-$sheet structure. The tertiary structure is the three-dimensional arrangement of amino-acids which is stabilized by hydrophobic interactions between the non-polar residues and through disulfide bonds in some proteins. Finally, quaternary structure is number (stoichiometry) and relative position of of two or more polypeptide chains held together by non-covalent bonds in multimeric proteins (Fig. 3.2). In this thesis, we use mass information derived from the primary structure of each protein in the heteromeric protein assemblies to study its quaternary structure. Also, in the sequel we refer the structure of macromolecular assemblies to their quaternary structure.

The two gold standards of the structural information are provided by x-ray crystallography and NMR spectroscopy. However, the low crystal quality for large macromolecular assemblies limits the use of the above high resolution techniques. This owes to the inherent flexibility and the relative sparsity of contacts with which to hold a crystal lattice together [Dyd10]. NMR spectroscopy is also limited by the sheer sample size. Therefore, one resorts to various experimental methods including biochemical and imaging methods to provide insights into the structure and dynamics of a large macromolecular assembly. These blocks of information put together provide a closer peek into the machinery of a macromolecular assembly.

The various experiments that are employed to give insights into different aspects of a large macromolecular assemblies are – cryo EM density maps, tandem affinity purification, mass spectrometry, tandem mass spectrometry (MS/MS), ion mobility mass spectrometry (IM-MS) etc.

The cryo-electron microscopy (EM) provides alternative way of visualization of large protein assemblies in discrete physiological and biochemical states. The problem is solved by fitting the individual protein shapes in their corresponding density regions of the map. The resolution of cryo-electron images is however low in most of the cases to provide detailed mechanistic description. Other limitation is that for those proteins for which homologous structures are not available, the quality is dependent upon manual assignment of the structures based on visual interpretation of density [LBC+08].

The tandem affinity purification (TAP) is a biochemical technique to determine protein-protein interactions. In this technique, list of binding partners in all the complexes containing a particular protein P of interest is determined. The protein P is engineered by introducing two sticky tags separated by a sequence corresponding to a protease cleavage site. Following two purification steps, a list of proteins of all the complexes containing the protein P is determined using mass spectrometry [PCR+01]. The limitation of this approach is in inference of connectivity and identification of complexes which can combinatorially grow.

Mass spectrometry (MS) has emerged as a key approach in structural biology. The recent advances have made it possible to transfer large macromolecular assemblies into the vacuum without their dissociation [Loo97], thereby allowing to measure the mass of an intact assembly. Measurements from MS and other modalities such as tandem mass spectrometry (MS/MS) and ion-mobility mass spectrometry (IM-MS) could be used to provide information about the stoichiometry of the subunits composing the protein assembly, inter-subunit interactions within the assembly, core and peripheral subunits in the assembly, insights into the size of an assembly etc. [Sha10].

In this thesis, we particularly focus on two of the above aspects, namely, stoichiometry determination of subunits and inter-subunit interactions within the assembly.

3.2 Mass Spectrometry: Overview

3.2.1 Motivation

Mass spectrometry (MS) in the last decade has been used extensively to probe the architecture of various protein assemblies [THS+08] [ZSF+08], [SMBE+09], [HDT+06]. It is now possible to measure the mass of intact assemblies by transferring them in the vacuum without dissociation with unprecedented precision and accuracy of the mass measurement. For illustration, when for several complexes, their masses calculated from the primary sequence is compared against that measured by the means of nano electrospray ionization mass spectrometry (nESI-MS), it is seen that there is a 1 : 1 correlation between the measured and the calculated mass with less than 1.5% discrepancy (Fig. 3.3 , chart 1) that arises due to the residual binding

of solvent and the buffer [BR11]. In addition, the MS can also be used to extract of relative population of a component in the mixture. For instance, when a sample of $\alpha\beta$-crystallin is measured by multi-angle light scattering couple to size-exclusion chromatography and compared with the measurement done by MS, the resulting distribution in the chart 2 of Fig. 3.3 suggests excellent agreement. It also suggests that the solution phase distribution of oligomers is maintained in the gas phase and hence quantifiable by MS [BR11].

Once the intact assembly has been transferred intact into the mass spectrometer, the analyst can manipulate it to provide information on the nature of the subunits and then interactions between the subunits. A technique called collision induced detection (CID) provides insights into whether a subunit is relatively exposed on the surface or buried into the core. Ion mobility (IM) can be integrated with the MS (IM-MS) to provide insights into the size of the assembly, based on the ability of ions to traverse a pressurized ion guide under the influence of a weak electric field. Both these, CID and IM-MS are described in detail in the later sections.

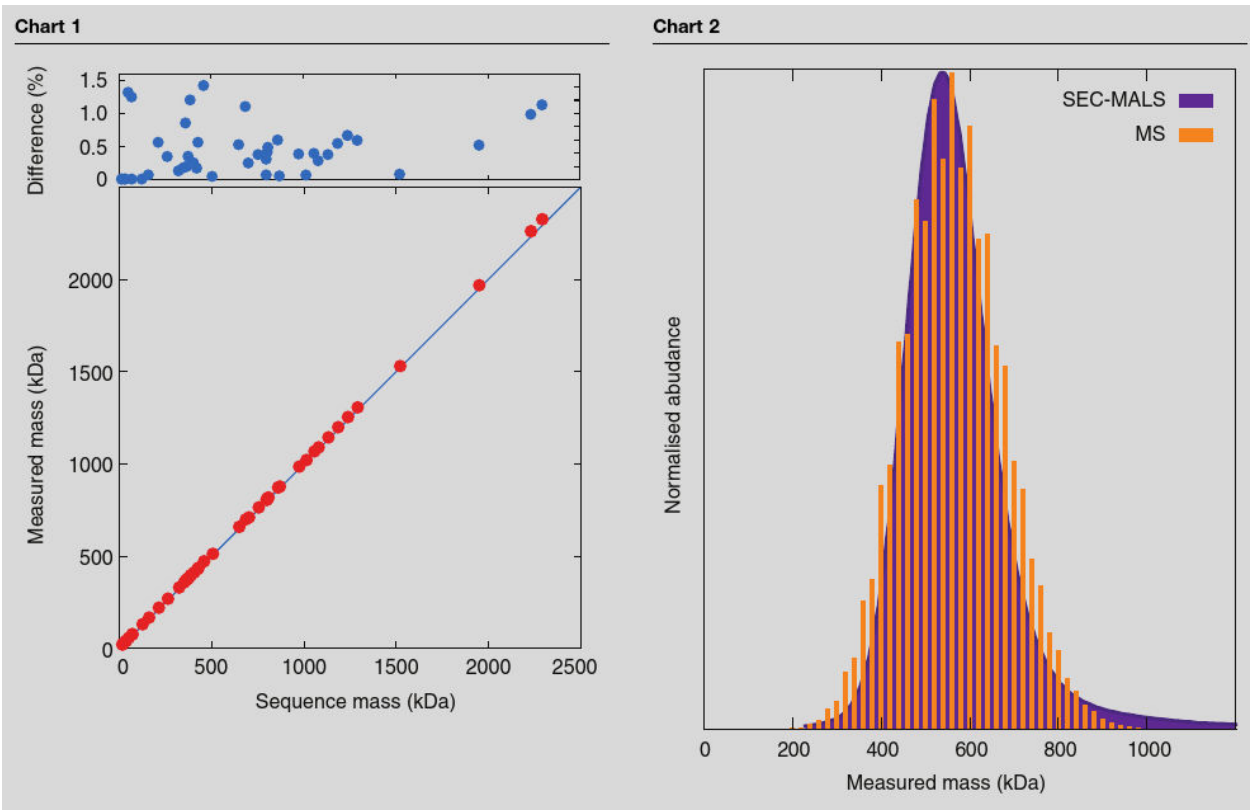


Figure 3.3: **Mass spectrometry measurement done on protein oligomers demonstrate efficacy, high accuracy and resolution.** [Chart 1] A comparison between measured mass (red dots) and mass calculated from sequences (blue line) of range of proteins and complexes. [Chart 2] A comparison of mass distribution of $\alpha\beta$-crystallin obtained by multi-angle light scattering coupled to size exclusion chromatography (SECS-MALS)(purple) and MS(orange).

3.2.2 Principles

General introduction. Mass spectrometry (MS) is an analytical technique allowing to determine the composition of a sample of material into its constituting *characters* (atoms or molecules), based on mass-to-charge (m/z) spectra of ions produced from the sample. The first m/z measurement was done for elementary particles called 'corpuscles', later known as electrons, by J. J. Thomson in 1897. The technique has undergone many developments ever since resulting in application of electron ionization mass spectrometry to the

peptides for the first time by K. Biemann in 1959 [BSG59]. In 1988, John Fenn and Masamichi Yamashita were successful in generating intact large biological molecular ions in gaseous phase at atmospheric pressure using a new ionization technique they developed called Electrospray Ionization, first reported in 1984 [YF84]. Following which, John Fenn was awarded Nobel prize in chemistry in 2002 along with Koichi Tanaka for his contributions in development of matrix-assisted laser desorption-ionisation (MALDI). In the last decade, then it has become an important analytical tool for study of large macromolecules, their assembly and disassembly in real time.

Main components of a mass spectrometer The mass spectrometer essentially consists of five main parts (Fig. 3.4): (1) *Ionisation source* to introduce the sample as an ion, to the mass spectrometer, (2) *a mass analyser* to sort the ions according to their m/z ratio, (3) *an ion detector* to detect the charged particle having a particular mass, and (4) *a data handling facility.*

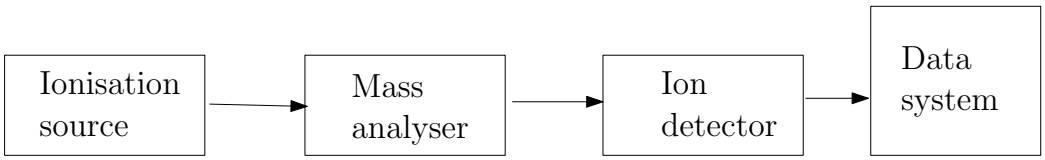


Figure 3.4: **Schematic diagram of a mass spectrometer**

Measures indicating effectiveness: Transmission efficiency, molecular mass accuracy and resolution. An important measure of the sensitivity of a mass spectrometer *Transmission efficiency* which refers to how many ions produced in the source region actually reach the detector. While studying protein assemblies held together by noncovalent interactions through Electrospray mass spectrometry (ESI-MS), the ion signal at the detector is proportional to the transmission efficiency [GRDP09].

The efficacy of the mass spectrometer is defined based on *mass resolution* and *molecular mass accuracy.* Mass resolution refers to the ability to separate two mass signals differing by a certain mass. A mass spectrometer separating two masses m and $m + \Delta m$ is said to have resolution,

$$R = \frac{m}{\Delta m}. \tag{3.1}$$

Molecular Mass Accuracy on the other hand, is the difference between the measured and theoretical masses for a certain ion. It can be expressed either in terms of percentage of the measured mass (e.g. molecular mass $= 10kDa \pm 0.01\%$) or as parts per million (molecular mass $= 10kDa \pm 100ppm$). It is to be noted that accuracy of measurement is linked to the resolving power of the instrument. Low resolution power instrument cannot provide high accuracy. Typical values of measurement accuracy and mass resolution are shown in Table 3.1.

Mass measurements: monoisotopic mass, average mass. Mass of a molecule can be expressed in many ways. *Nominal mass* is calculated by using mass number of the most abundant isotope, without regard of mass defect/excess (e.g. $H = 1$, $C = 12$, $N = 14$, $O = 16$, etc.). *Monoisotopic mass* refers to the sum of masses of elements corresponding to their most abundant (stable) isotope (i.e., $^1H = 1.007825Da$, $^{12}C = 12.000000Da$, etc.). However, *Average mass* is the sum of the average mass of elements, their stable isotopes being weighted for abundance (e.g., 98.9% for ^{12}C and 1.1% for ^{13}C etc.)

Relationship w.r.t. to small systems and large systems. Whether the monoisotopic mass or average mass is to be used while reporting depends on the mass of the molecule and the resolving power of the mass spectrometer. From Fig. 3.5, it can be seen that with increase in the mass of the molecule the relative

abundance of monoisotopic mass decreases. In addition, the isotopic distribution resembles that of a normal distribution. This observation is principally due to presence of a relatively abundant heavy isotope. Proteins have high content of carbon and hydrogen and therefore the difference between monoisotopic mass and the average mass is amplified as mass of the molecule increases [Str05].

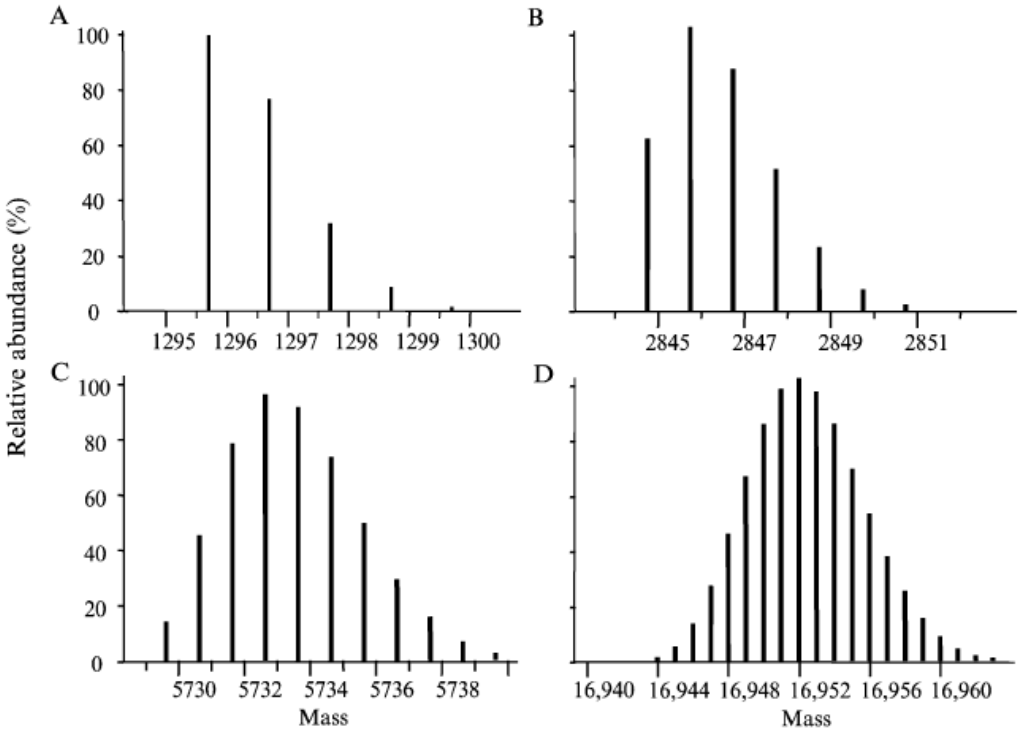


Figure 3.5: **Isotopic distribution of masses (in Da) for four systems of varying molecular mass.** (A) Angiotensin I, human ($M_{MONO} = 1295.6775\,Da$, $C_{62}H_{89}N_{17}O_{14}$). (B) Melittin, honeybee ($M_{MONO} = 2844.7542\,Da$, $C_{131}H_{229}N_{39}O_{31}$). (C) Insulin, bovine ($M_{MONO} = 5729.6009\,Da$, $C_{254}H_{377}N_{65}O_{75}S_6$). (D) Apomyoglobin, horse ($M_{MONO} = 16940.9650\,Da$, $C_{769}H_{1212}N_{210}O_{218}S_2$). Picture from [Str05]

3.2.3 Instruments

Ionisation techniques

There are a number of different methods available, however, for the study of protein complexes only two are relevant namely Electrospray ionisation (ESI) and Matrix assisted laser desorption ionisation (MALDI). Both of them are soft-ionisation techniques.

Matrix-assisted laser desorption ionisation (MALDI). In this technique, a high power density laser beam is focused on a small spot of the sample mixed with the matrix resulting in extremely high heating. The matrix is commonly a UV absorbing chromophore normally an organic acid. This forms a plume of vaporised molecules both of the analyte and the matrix. The ions are formed by proton transfer between excited molecules of the matrix and the sample molecules and also by the collision cascade in the expanding plume (Fig. 3.6). The use of matrix is central in MALDI for it absorbs the laser radiation increasing the lifespan of the sample. In addition, the use of matrix prevents extensive fragmentation. The molecular weight of the molecules that can be analysed by MALDI can range upto 1000 kDa (i.e., large protein assemblies) [CGH$^+$13].

Electrospray ionisation (ESI). Whilst some research groups have used MALDI to study large non-covalent assemblies, the vast majority of work uses ESI. In ESI, ions are generated from the sample solution at atmospheric pressure. The sample solution is passed through a narrow capillary tube at a low flow rate (1-20 μl/min). The high electric potential applied between the capillary and a counter electrode nearby causes the liquid to disperse into a fine spray of charged droplets. The polarity of the charged droplets produced depends on the polarity of the voltage applied to the capillary with respect to the entrance to the mass spectrometer. These highly charged droplets undergo nebulization process facilitated by the source temperature and flow of N_2 gas. The radius of the droplets decreases due to evaporation. The decrease in the radius of a droplet results in increase in surface charge density. Small highly charged ions eject from the surface once the electric field strength reaches a critical size which is Rayleigh limit. These emitted ions are accelerated to the mass analyzer for further analysis (Fig. 3.7). ESI produces multiply charged ions of the general form $[M + \mathrm{n}H]^{nZ}$ from which charge (nZ) and mass of the analyte (M) is computed (Fig. 3.8). One striking thing about this technique is the possibility of ions being multiply charged. The number of attached protons to a peptide or protein correlates well with basic amino acids (Arg, Lys, His) plus the N-terminal amino group, unless it is acylated along with pH, temperature and any denaturing agent present in the solution. This bags in several advantages for ESI. Firstly, due to the multiple charges, high mass molecular ions can be analysed at low m/z range (Fig. 3.8). Secondly, high resolution detection of large mass ions is possible as the mass resolving power is inversely proportional to m/z. Thirdly, it is one of the most gently ionization technique yielding no fragmentation, which makes it suitable for large protein assemblies.

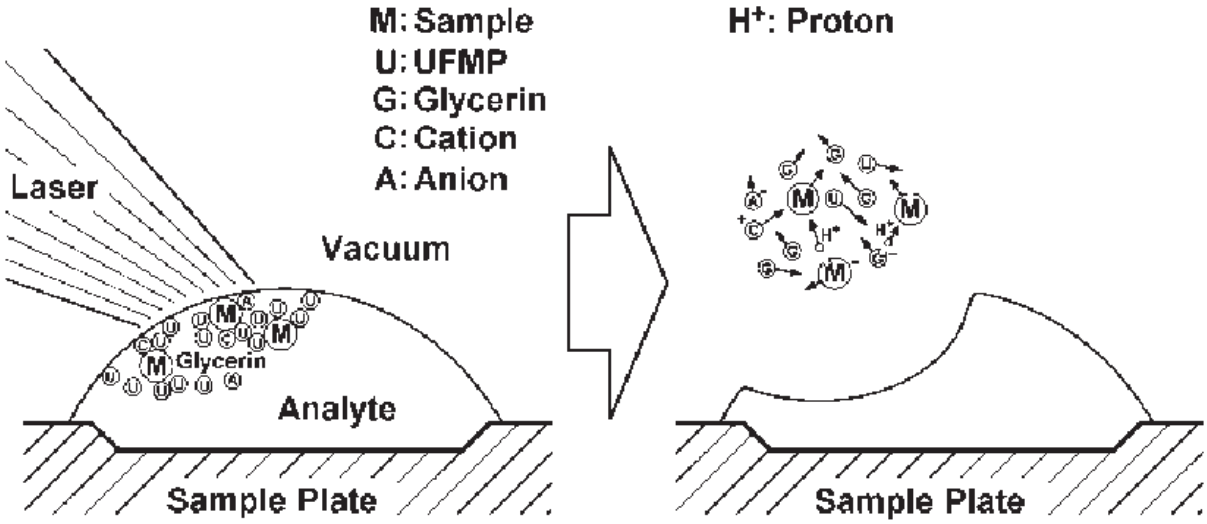


Figure 3.6: **Mechanisms of ionisation: MALDI** Picture from K. Tanaka Nobel lecture 2002.

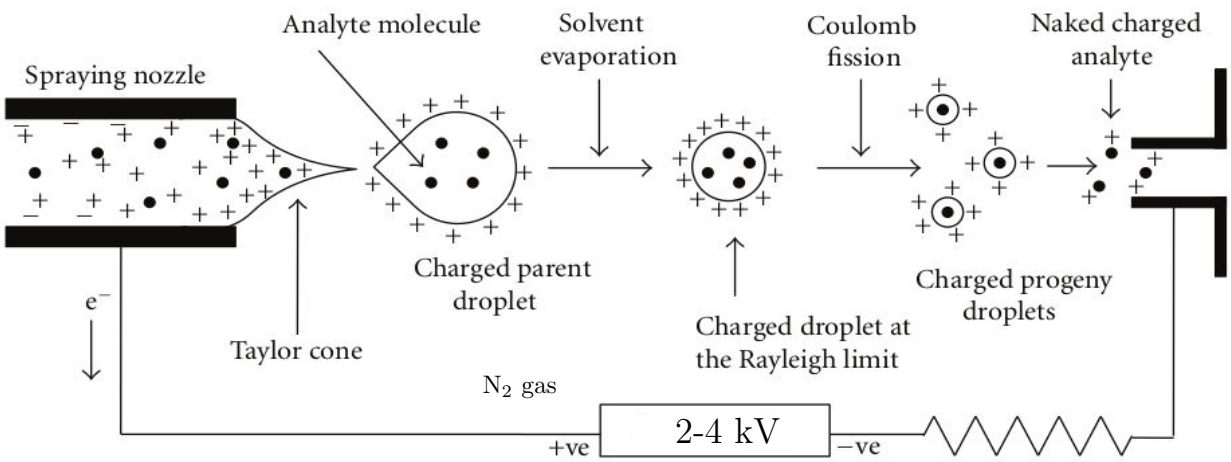


Figure 3.7: **Mechanisms of ionisation: electrospray ionisation** Modified from picture from [BM12].

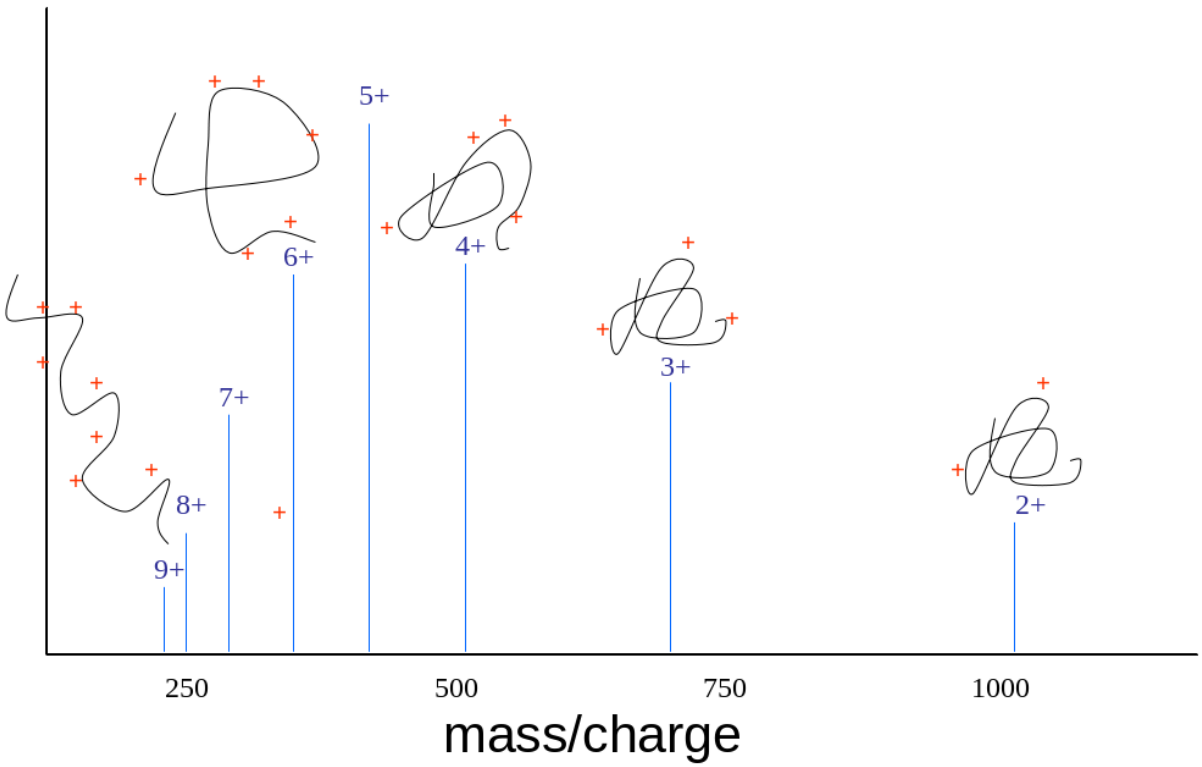


Figure 3.8: **Typical charge distribution from an ESI spectrum.**

Mass analysers

The ions formed during the ionisation process are then separated by mass analyzers. Broadly speaking, there are three types of mass analysers: Scanning analyzers, time of flight (TOF) analyzers and trapped-ion analysers.

Scanning analyzers include magnetic sector instrument and quadrapole analyzers. In trapped-ion analysers the ions are trapped inside the analyser. Examples include quadrapole ion trap, fourier-transform ion-cyclotron resonance (FT-ICR), orbitrap mass analyzer etc. For the purpose of this thesis we will only describe the two types of analysers used for native mass spectrometry - quadrapole time-of-flight (Q-TOF) and orbitrap (for extended mass range) analyzers.

Quadrapole mass analyser. A Quadrupole is a mass analyzer that uses an electric field to separate ions. The Quadrupole consists of 4 parallel rods/ poles, where adjacent rods have opposite voltage polarity applied to them. The voltage applied to each rod is the summation of a constant DC voltage (U) and a varying radio frequency ($V_{rf}\cos(\omega t)$), where ω stand for the angular frequency of the radio frequency field. The electric force on the ions causes the ions to oscillate/orbit in the area between the 4 rods, where the radius of the orbit is held constant.

The ion moves in a very complex motion that is directly proportional to the mass of the ion, voltage on the quadrupole, and the radio frequency. The ions will remain orbiting in the area between the poles with no translation along the length of the poles unless the ions have a constant velocity that is created as the ions enter the quadrupole. Before entering the analyzer, the ions travel through a potential of a certain voltage, usually created by ring electrode, in order to give the ions a constant velocity so they can transverse along the center of the quadrupole.

While in the quadrupole, the trajectories of the ions change slightly based on their masses. Ions of specific mass have a certain frequency by which they oscillate. The greater the mass, the greater the frequency. A certain limit is associated with each quadrupole and it selects ions which are within the desirable frequency range

Time of flight (TOF). It is based on the principle that ions with different m/z value have different velocities and therefore reach the ion detector at different times. The ions having charge z, mass m accelerated through the electric potential V_{acc} have kinetic energy zV_{acc}, independent of mass. Equating the accelerating potential to the kinetic energy of ions, their velocity is given by

$$v = (\frac{2zV_{acc}}{m})^{1/2}. \tag{3.2}$$

The time t lapsed during the flight in traversing distance L before reaching the detector is given by,

$$t = (\frac{m}{2zV_{acc}})^{1/2}L \tag{3.3}$$

Thus by measuring the time, t and knowing the electric potential V_{acc} and length, L of the analysing tube m/z of an ion can be determined. The typical electric potential the ions are subjected to is 1-20 kV and length of the tube, typically 0.5-2.0 meters. The advantages of TOF include, theoretically unlimited mass range, high transmission (most of the ions injected are detected), high speed of measurement of the order of microseconds. However, the technique has low resolution. Typical resolving power of TOF instruments is not greater than 1000 [SZZ07].

More commonly, quadrapole and TOF analysers are modified and clubbed together to form Q-TOF analysers (Fig. 3.9). Most of the modifications are made to improve transmission of high mass ions. In the source region the operating pressure is increased to 10 millibars from 1 millibar. This focuses high-mass ions through collisional cooling. The transmission efficiency is further improved by use of heavier buffer molecules, e.g., argon or xenon instead of helium or nitrogen. The collision cell is also operated at high pressure and in the presence of heavier inert collision gas molecules (xenon, kryton). The collisions with heavier inert gas molecules strip ions or buffer adducts from the analyte resulting in improvement in spectral quality. If the energy of collision is sufficiently higher then gas phase dissociation yielding fragment ions which can help confirm mass assignment and also provide topological and positional constraints. More on collision induced dissociation (CID) is discussed in the section 3.2.5.

	TOF	Quadrapole
Mass limit (m/z)	$> 10^6$ Th	4000 Th
Mass accuracy	200 ppm	100 ppm
Resolution, $(m/\Delta m)$	5000	2000

Table 3.1: **Comparison of mass analysers**. The definitions of units are as follows: 1 Thomson (Th) = 1 $u/e = 1.036426 \times 10^{-8}\, kgC^{-1}$. Parts per million (ppm) is a unit to measure concentration of very dilution solutions. The concentration of 1 ppm is equal to 1 part of the solute per 1 million parts of the solution. Table from [ES07].

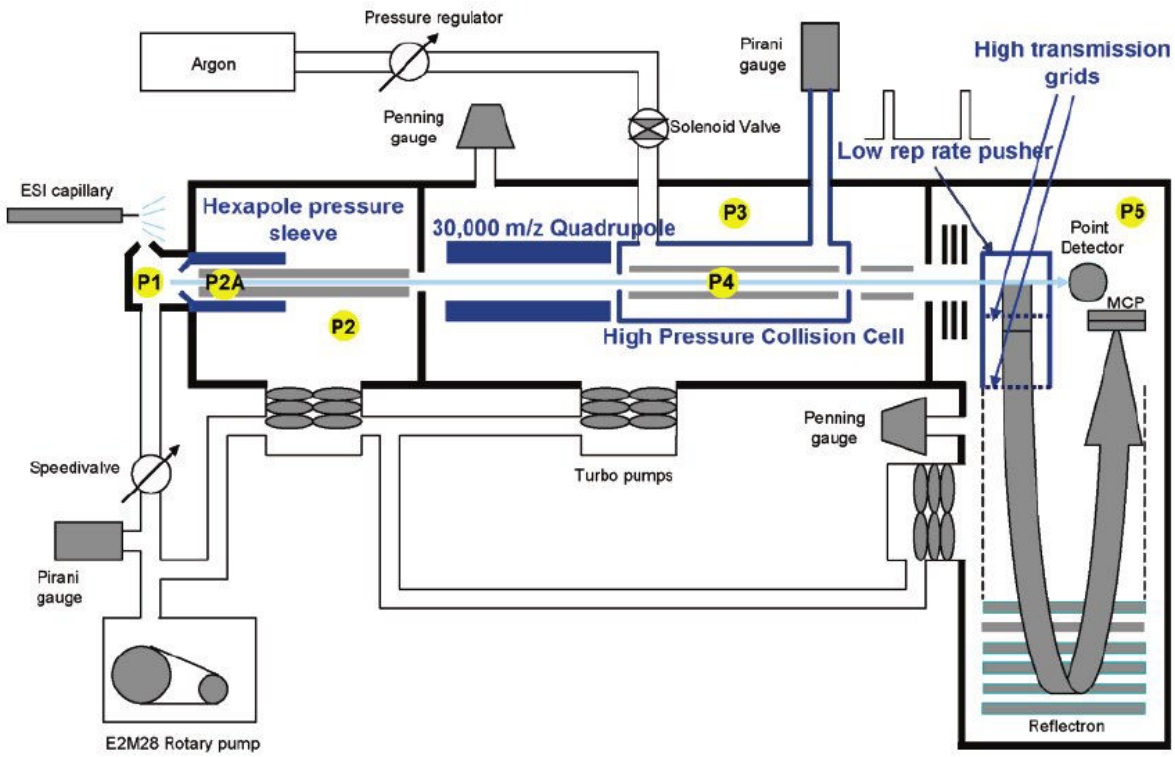


Figure 3.9: **Schematic layout of first generation Q-TOF instrument.** Picture from [vdHvDM+06].

Orbitrap mass analyzer. It is an ion trap mass analyzer composed of a spindle like central electrode and a barrel-like outer electrode (Fig. 3.10). The idea is to confine the rotational ion motion first proposed by Kingdon in 1923 by trapping the positive ions around a negatively charged filament [Kin23]. This approach was further developed by Knight who proposed a quadro-logarithmic field ([Kni81]) being produced between specially shaped electrodes (Fig. 3.10) shown in the Eq. 3.4 below:

$$U(r,z) = \frac{k}{2}(z^2 - \frac{r^2}{2}) + \frac{k}{2}(R_m)^2 \log \frac{r}{R_m} + C \tag{3.4}$$

where, r and z[2] are cylindrical co-ordinates, R_m is the characteristic radius, k is the axial restoring force and C is a constant.

The ions are injected at an offset from the equator ($z = 0$) perpendicular to $z-$axis as shown in the Fig. 3.10. The ions not only orbit around the central electrode in elliptical paths but also oscillate simultaneously

[2]The alphabet z while describing orbitrap analyzer is reserved for the axial direction. However, elsewhere in this thesis it would mean charge on the ion, unless otherwise stated.

along the z direction, that is why their trajectories look like helices. The equations of motion under the influence of a quadro-logarithmic field is described in detail by Makarov [Mak00]. Those ions with orbital radius less than R_m are trapped. Also, the electrostatic potential produced between the two axially symmetric electrodes from the Eq. 3.4 is such that the motion of ions in z direction is independent of their motion around the central electrode and also of initial parameters of ions except their mass-to-charge ratio. Its axial frequency is given by:

$$\omega = (\frac{kq}{m})^{1/2} \tag{3.5}$$

where, m and q represent mass and charge respectively of an ion and k is the force constant of the potential applied. The Eq. 3.5 shows that the frequency of oscillations along axial direction (z direction) is only dependent upon the mass-to-charge ratio of an ion and the potential (kept constant) between the electrodes.

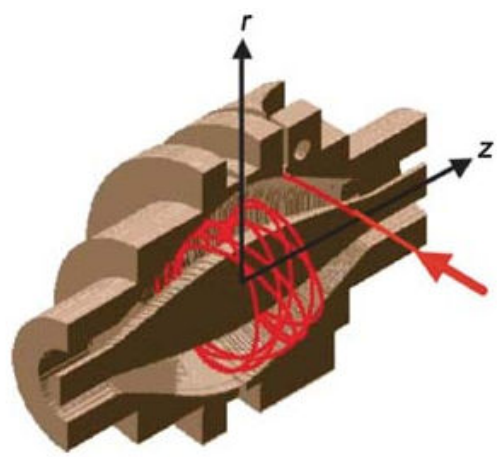


Figure 3.10: **Cutaway view of the orbitrap analyzer.** The red arrow in the figure is where the ions are injected. Picture from [PCN08].

Ion detection

The ions that pass through the mass analyser are segregated according to their m/z ratio. They enter into the detection chamber and are electrically detected by a detector. Some of the detectors used are Electron Multiplier, Faraday cup. In case of an Electron Multiplier, when an ion hits the detector surface the secondary electron emission occurs from the atoms of the surface. These emitted electrons are accelerated to another metallic plate resulting in another secondary emission. This process is continued to amplify the signal. Faraday cup is based on the principle that when a beam of ions hits the metal its gains small net charge from the ions. The metal is then discharged to measure the small current from which number of ions initially hit can be determined.

3.2.4 Native Mass Spectrometry

Native MS refers to the mass analysis of protein complexes under nondenaturing conditions so as to preserve their noncovalent interactions. The buffered aqueous solution is used to mimic the physiological conditions still compatible with ESI. ESI is one of the most gentle ionization methods. If the flow rates used in the capillary further lowered then the size of the droplets formed is smaller. The number of fission events are also lowered before effective ionization of an analyte making the process even more gentler. Such, electrospray ionization method if the flow rate in the capillary is scaled down to few tens of nanolitres per minute is called *nano-ESI (nESI)* and is the preferred ionization method for MS of protein complexes [SH14].

In the case of noncovalent protein complexes, the buffer adduct formation is likely resulting in broadening of peaks for higher masses. The broader peaks provide difficulty in resolving the charge state of corresponding

peaks as they overlap extensively. In such cases the resolving power of an instrument could be enhanced by improving the desolvation of ions.

Rose et al. [RDD+12] demonstrated the use of orbitrap analyzer to measure the mass of protein assemblies with a mass accuracy of 10 ppm. This is due to the improvement in desolvation efficiency which is achieved by allowing the ions to pass through the high-energy collision induced dissociation (HCD) cell before their entrapment in the C-trap. This increased desolvation, allowed trapping of large protein complexes and drastically improved the sensitivity. The schematic of the modified instrument is shown in the Fig. 3.11.

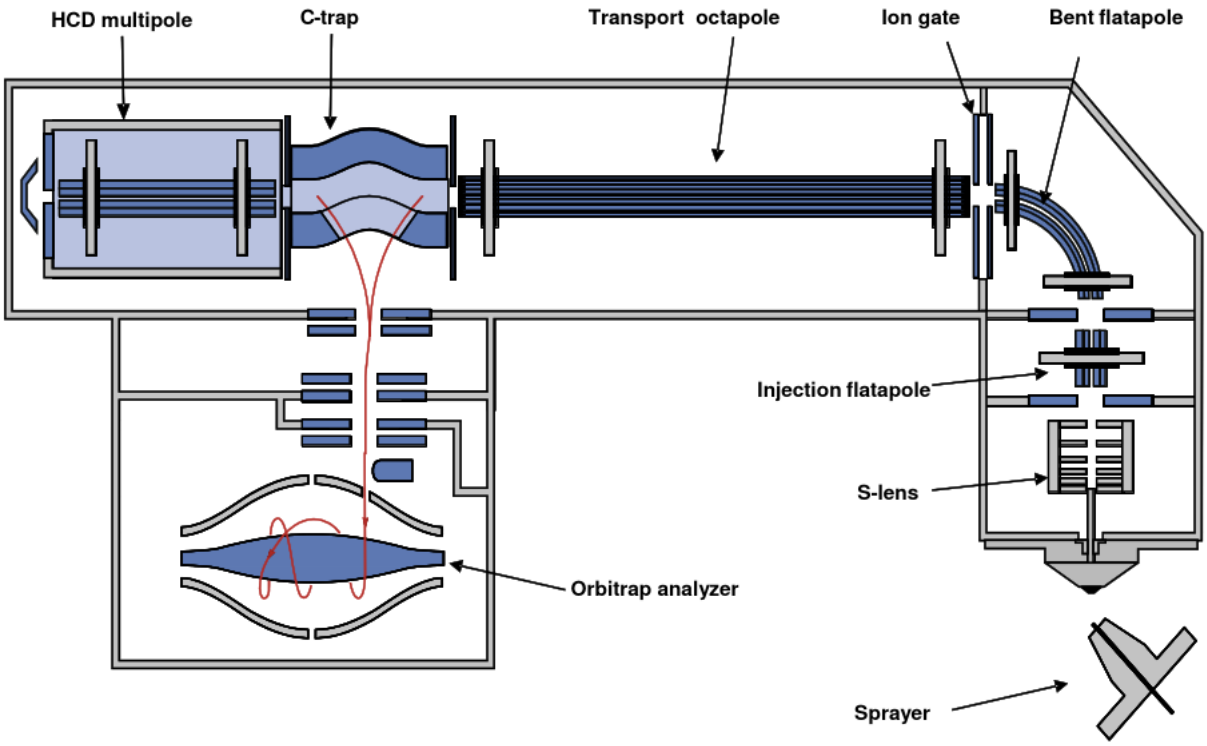


Figure 3.11: **Schematic of modified mass spectrometer with orbitrap mass analyzer customized for protein assemblies.** Picture from [RDD+12].

3.2.5 Tandem Mass Spectrometry (MS/MS)

General principle

A technique in a mass spectrometer which consists of two or more analysers usually hyphenated together, e.g. Q-TOF (see section 3.2.3) used to analyse the structure of biological molecules is tandem mass spectrometry (MS/MS). It involves two stages of MS. The 1st MS is performed on a single m/z. This ion called parent ions or a precursor ion, is passed into a collision cell where they undergo dissociation as a result of conversion of kinetic energy into the internal energy of the ions. The dissociated ions, called the product ions are then analyzed by the second stage of MS.

Methods of dissociation. The dissociation can be realized through various mechanisms such that collision induced dissociation (CID) which is the most frequently used technique. Other techniques include surface induced dissociation (SID) and photodissociation (Fig. 3.12). In CID, the parent ions are made to collide with inert gases such as argon, helium of xenon, where the kinetic energy of the selective ions is converted into internal energy resulting into dissociation of ions. In SID, the collision of parent ions is realized with a

surface modified with a self assembled monolayer. Finally, in photodissociation, the photons are focused on the the parent ions resulting in dissociation. With techniques such as infrared multiphoton photondissociation (IRMPD) implemented on trapping instruments, the parent ions can be maintained in the laser beam for extended periods of time, allowing absorption of multiple photons [LSS+94].

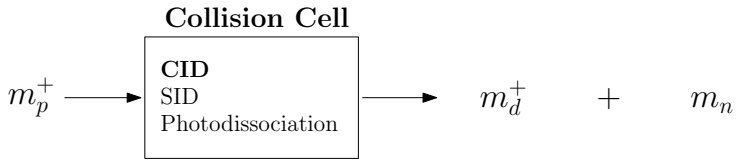


Figure 3.12: **Dissociation i.e. framentation of ions through various mechanisms in tandem mass spectrometry.** m_p^+, m_d^+, m_n are respectively, parent ion, product ion and the neutral fragment. Note that m_n can also be charged if the ionisation by ESI yields multiply charges ions.

CID on multiprotein complexes mostly produces a highly charged unfolded monomer and complimentary (n-1)mer product ion regardless of the size, composition and architecture of the precursor complex. On the other hand, SID yield charge-symmetric product ions, implying less structural change prior to the dissociation. Blackwell et al. [BDBW11] have demonstrated the nature of dissociation of protein assemblies through CID and SID on an heterohexamer, toyocamycin nitrile hydratase. It is observed that CID yields a highly charged unfolded monomer and a pentamer whereas, SID produces its constituent trimers (Fig. 3.13). The direct observation of the representative substructure through SID reveals its potential in structural biology.

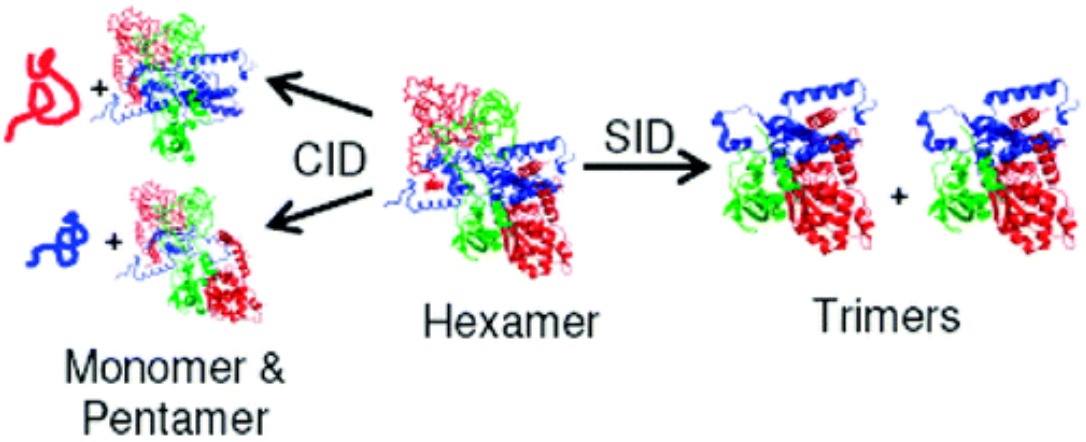


Figure 3.13: **Comparison of CID and SID on an heterohexamer, toyocamycin nitrile hydratase.** CID yields asymmetric charged oligomers of different size whereas, SID yields charge-symmetric product ions.

3.2.6 Ion Mobility Mass Spectrometry

Ion-mobility spectrometry (IMS) is an analytical technique used to separate and identify ionized molecules in the gas phase based on their mobility in a carrier buffer gas. Though heavily employed for military or security purposes, such as detecting drugs and explosives, the technique also has many laboratory analytical applications, recently being coupled with mass spectrometry.

In principle, ions trajectory in a weak electric field is hindered by the neutrally charged molecules present in the MS environment [MM88]. The collision with the neutral molecules slows down the trajectory of ions. The measured drift time (t_D) is proportional to the cross-sectional area (Ω) of an ion and can be related

to its quaternary structure. In addition, another piece of information, Ω, coming from IM-MS aides in studying two different complexes having similar m/z values. The schematic diagrams for two contemporary time-dispersive ion mobility mass spectrometer instrumentation are shown in the Fig. (3.14).

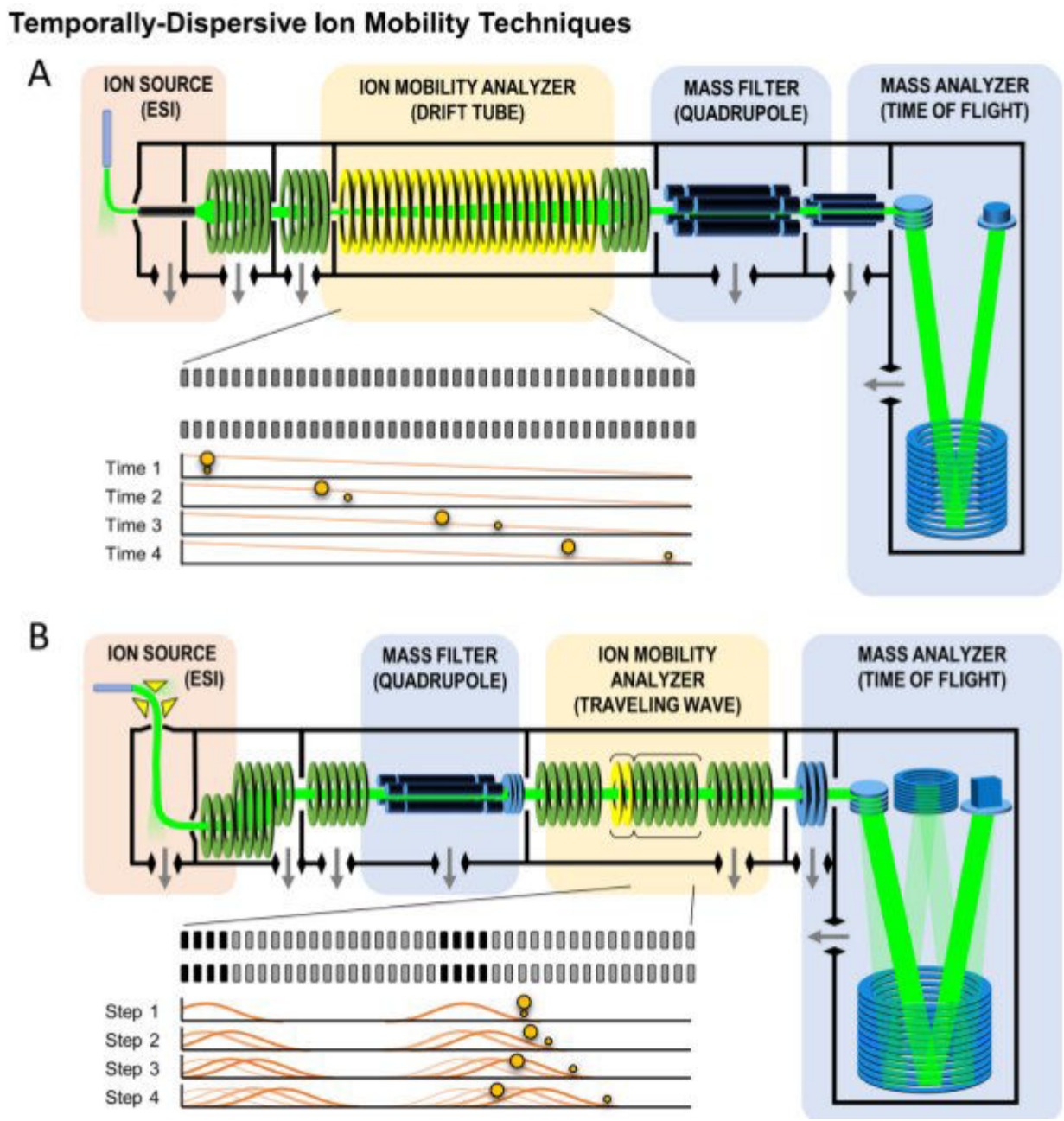


Figure 3.14: **Schematic diagrams of Ion mobility Mass Spectrometer instrumentation.** (A) A drift tube ion mobility spectrometer (DTIMS); (B) A traveling wave ion mobility spectrometer (TWIMS). Picture from [MM15].

Ion mobility mass spectrometry (IM-MS) are not explicitly considered in this thesis nonetheless this will impact future studies.

3.3 Molecular systems handled: from chemistry to structural biology

Metabolites

Mass spectrometry proved instrumental in metabolomics, which is the comprehensive and quantitative study of sum total of metabolites found in cells, tissues, organ or organism [DAH07]. Identifying a small molecule consists of finding its elemental composition.The monoisotopic masses of the compounds are in general not sufficient to do so, so that isotope patterns are also used [KF06]. First all the possible organic formulas are determined from the monoisotopic mass and then for each candidate organic formula, its isotopic pattern is simulated and matched with the input pattern to find the right candidate [BLLP09]. Typical mass range of metabolites in the KEGG database [3] is < 1000 Da.

Proteins

Mass spectrometry is widely used for characterization of proteins. Broadly speaking, there are two ways of studying a protein. The first, which is called "top-down" strategy includes ionization of intact proteins which are then sent to the mass analyser. The second involves digestion of protein in question by the restriction enzymes to yield peptides. These peptides are further analysed to identify the source protein. This approach is called "bottom-up" approach. Following proteolysis, the sequencing of a each peptide can also be done based on their masses. This is called De novo sequencing. In order to account for different amino acids having the same mass, one can use sequence homology in addition to the database search and de-novo sequencing.

Protein assemblies

More recently, mass spectrometry also proved invaluable to elucidate the stoichiometry of proteins involved in complexes varying in mass, size, solubility, and bound/unbound states [SR07, BR11]. Typical mass of a protein complex composed of around 10 proteins can weigh several 100 kDa, e.g., Yeast RNA exosome, 400 kDa; 26S proteasome, 2.5 MDa etc.

Building models of macro-molecular machines is a key endeavor of biophysics, as such models not only unravel fundamental mechanisms of life, but also offer the possibility to monitor and to fix defaulting systems. Example of such machines are the eukaryotic initiation factors which initiate protein synthesis by the ribosome, the ribosome which performs the synthesis of a polypeptide chain encoded in a messenger RNA derived from a gene, chaperonins which help proteins to adopt their 3D structure, the proteasome which carries out the elimination of damaged or misfolded proteins, etc. These macro-molecular assemblies involve from tens to hundreds of molecules, and range in size from a few tens of Angstroms (the size of one atom) up to 100 nanometers.

Understanding the structure of these machines at the atomic level is a fundamental endeavor, as it allows unraveling the relationship between the structure and the function of bio-molecules, a pre-requisite to monitor such systems. But if atomic resolution models of small assemblies are typically obtained with X-ray crystallography and/or nuclear magnetic resonance, large assemblies are not, in general, amenable to such studies. Instead, their reconstruction by *data integration* requires mixing a panel of complementary experimental data [AFK+08]. In particular, information on the hierarchical structure of an assembly, namely its decomposition into sub-complexes (complexes for short in the sequel) which themselves decompose into isolated molecules (proteins or nucleic acids) can be obtained from mass spectrometry.

[3] http://www.genome.jp/kegg/

3.4 Data Processing

3.4.1 Obtaining Mass Spectra

While handling the data acquired from the mass spectrometry measurements, given below are the algorithmic challenges that are faced. They can also be used as guiding principles while designing an automated pipeline and criteria to compare two software solutions.

PRE-PROCESSING. For illustration purpose, the typical mass spectrum peaks are as seen in the Fig. 3.15. The data acquired from the mass spectrometer is pre-processed - Noise Rejection, Data Reduction followed by Feature Detection. They are explained in detail.

Noise Rejection. It is necessary to remove the noise hampering the signal detection even during the use of high-resolution mass spectrometers. There are three types of noises – Random Noise, Chemical Noise and Noise due to the presence of contaminants. Random noise is mainly of electrical origin represented by small spikes uniformly distributed in mass spectra. Chemical noise is introduced during the simultaneous detection of buffers by mass analysers. Finally the non protein contaminants such as plasticizers, surface contaminants pollute the signal and need to be removed or identified or ignored in processing.

Data Reduction. Raw data retrieved directly from the instrument, owing to the size, is inconvenient to handle by downstream algorithms. Therefore, it is possible to reduce to more manageable set of peaks such as by centroiding the MS spectra thereby retaining a single representative peak at the centre of m/z ion distribution measured by the instrument detector. This is not commonly employed in native MS.

Feature Detection. Due to the phenomenon of isotopic dispersion, a peptide is seen as a collection of peaks at different m/z values. The recognition of the isotopic pattern is a necessary step before abundance measurements. *Deisotoping* methodology can be used to detect features. Area under the curve (AUC) [4], of the plot of relative abundance vs m/z spectra, is computed for each monoisotope and summed for all isotopic peaks in a given scan. Averaging the sum over total elution time would give an estimate of feature volume. The limitation of using whole isotopic profiles is to make it vulnerable to contamination from co-eluting isobaric compounds. This is pertinent to Liquid chromatography mass spectrometry (LC-MS) datasets. On the contrary, the sensitivity of using only most abundant monoisotopic peak for feature detection is that sensitivity at higher masses is sufficiently low, e.g. the monoisotopic peak is 5 % of the total abundance at 5000 Da.

Upon determination of the masses of the ions from the m/z spectrum, depending on the nature of system under scrutiny, whether, metabolites, peptides or large intact protein assemblies, they undergo different treatment for further analysis. The specific pipelines employed for such systems are described below:

3.4.2 Specific pipeline for Metabolites

The ions of the small metabolites are post-processed and undergo following steps described below [NB10].

Simulating isotope patterns. As the monoisotopic mass of a compound is insufficient to determine its molecular formula, we can use the measured isotope pattern of the compound to rank all remaining molecular formula candidates. Due to limited resolution of most MS instruments the iso- topic variants are not fully separated in the spectra but pooled in mass bins of approximately 1 Da length. This is called the aggregated isotopic distribution and in the following we will refer to it as *isotope pattern*. Most elements have several naturally occurring iso- topes. Combining elements into a molecular formula also means to combine their isotope distributions into an isotope distribution of the entire compound. To this end, we can simulate the theoretical isotope pattern of a molecular formula, and compare the simulated distribution to the measured pattern of a compound.

[4] Not to be confused with the AUC used when studying *Receiver Operating Characteristic* curves.

Scoring candidates compounds by comparing isotope patterns. Decomposing the monoisotopic peak can result in a large number of candidate molecular formulas that are within the measured mass. We can rank these candidates based on evaluating their simulated isotope patterns. For each candidate molecular formula, the isotope distribution is simulated and compared with the measured one. The best matching formula is considered to be the correct molecular formula of the compound.

Isotopic labeling. Labeling compounds by isotope-enriched elements such as 13 C or 15 N , helps to identify the correct molecular formula. The shift in the mass spectrum between the unlabeled compound and the labeled compound indicates the number of atoms in the compounds. Once the number of atoms for the labeled elements is known, the number of possible molecular formula is significantly reduced. Rodgers et al [RBHM00] showed that enrichment with 99% 13 C isotopes reduces the number of possible molecular formu- las for a 851 Da phospholipid from 394 to one. Hegeman et al. [HSC$^+$07] used isotopic labeling for metabolite identifica- tion. They improved the discriminating power by labeling with 13 C and 15 N isotopes. Giavalisco et al. [GLM$^+$11] additionally labeled compounds with 34 S isotopes. By this, the number of carbon, nitrogen as well as sulfur atoms can be determined upfront, and the number of potential molecular formula that we have to consider, is reduced considerably. Baran et al [BBB$^+$10] applied this approach to untargeted metabolite profiling and showed its potential to uniquely identify molecular formulas.

3.4.3 Specific pipeline for Proteins: Bottom-up approach

In order to analyze the proteins by mass spectrometry, typically, proteins are digested by restriction enzymes such as trypsin yielding peptides. These peptides are then separated by liquid chromatography, converted into the gas phase and analysed by mass spectrometry. Following are the steps that peptides undergo after having obtained the mass spectra.

Peptide Identification . The strategies used to infer the peptides can be broadly categorized into three categories – Database searching, Library search methods and De novo sequencing. *Database search* is the most common among the three approaches. The observed mass spectra of a peptide is compared against all the simulated mass spectra of the peptides in the database. *Library search* method involves searching the signature of the observed spectra within the spectral library. Although, this methodology outperform the previous *database search* in terms of speed, error rates and sensitivity but this approach necessarily requires the spectral library to be exhaustive. The third approach, *de novo sequencing* is to interpret the amino-acid sequence of the peptide whose spectra is under scrutiny. Owing to the fact that this is computational intensive, this approach is used for unidentified high-quality spectra.

The peptide identification can become challenging taking post-translational modifications (PTMs) into account due to exponential explosion of the possible number of states of the peptide depending on the number of modification sites.

False Discovery Rates. One needs to determine the false discovery rate while identifying the peptides as it would have incidence on protein quantification subsequently. "Target-decoy" approach ([MYL02]) involves adding randomly generated sequences to the source database. *Database search* method is used now on this extended database to return fraction of false positives.

The above steps are concerned with the characterization of peptides in the sample. Peptides are involved in many regulatory processes and act as signal molecules, rendering peptidomics an important branch of study. However, depending on the problem and context, proteins identification and quantification are sought.

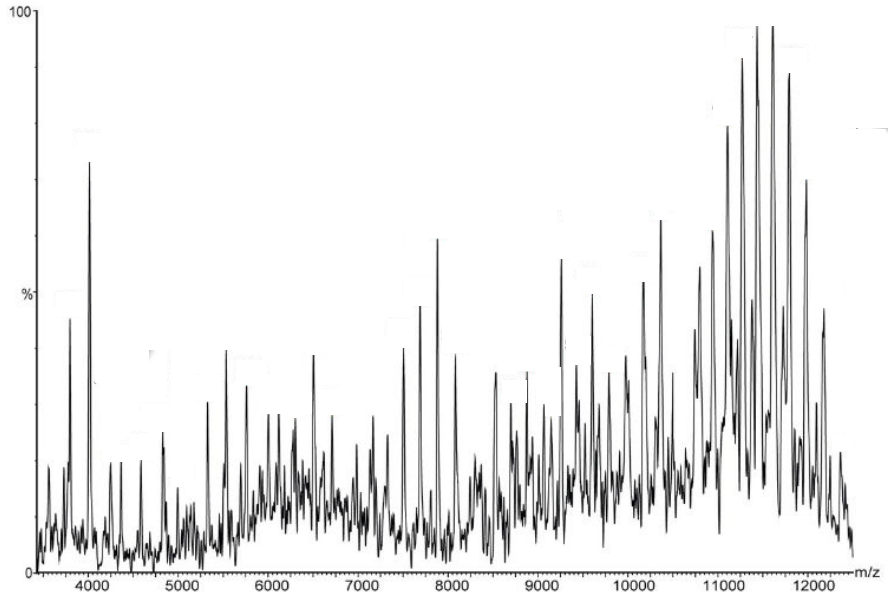
Protein Inference . After having characterized overlapping peptides, generated using a set of digestion enzymes, they are then assembled to infer the source protein. The problem is, however, ill posed. The overlapping peptides do not always map to a unique protein, instead, a subset of peptides is common to many proteins [QA10] present in the database. This non-uniqueness in the protein database can arise due to following reasons: (a) There could be a hundred of entries resulting from the expression of the same gene

because of splicing variants, post-translational modifications, protein isoforms and homologous proteins from other species.
(b) It could be due the processing a truncated protein that has a signature in another protein having a similar domain. Also, if the length of the peptide is very small they become promiscuous thereby posing difficulty in unique mapping.
(c) Multiple entries of the protein could also be due to sequencing errors.
False discovery rate can also be calculated for protein inference based on a certain criteria. For example, a threshold of minimum number of peptides mapped could be set for a candidate protein to qualify as been identified ([CAB+04]).

Protein Quantification . For a given protein, after peptide quantification and protein inference step, two ways could be used for protein quantification. The first is to calculate different ratios from the protein's peptides, followed by summarizing these ratios to obtain a single-fold change. This method is more popular with stable-isotopic labeling. Furthermore, the standard deviation of the protein ratio can be derived from the peptide ratio. The second method is to deduce estimate of protein abundance from its peptides, followed by determining a single fold change at the protein level.
The spectral counts[5], owing to the empirical relationship with protein abundance, can be used to determine the absolute concentration of each protein within a mixture. The normalization procedures yield absolute concentration by correcting for differing propensities of proteins to produce identifiable fragmentation spectra.

[5]The spectral count is the number of MS/MS spectra identified as arising from a certain species. The rationale is that the fragmentation events are proportional to the abundance of the protein.

A.



B.

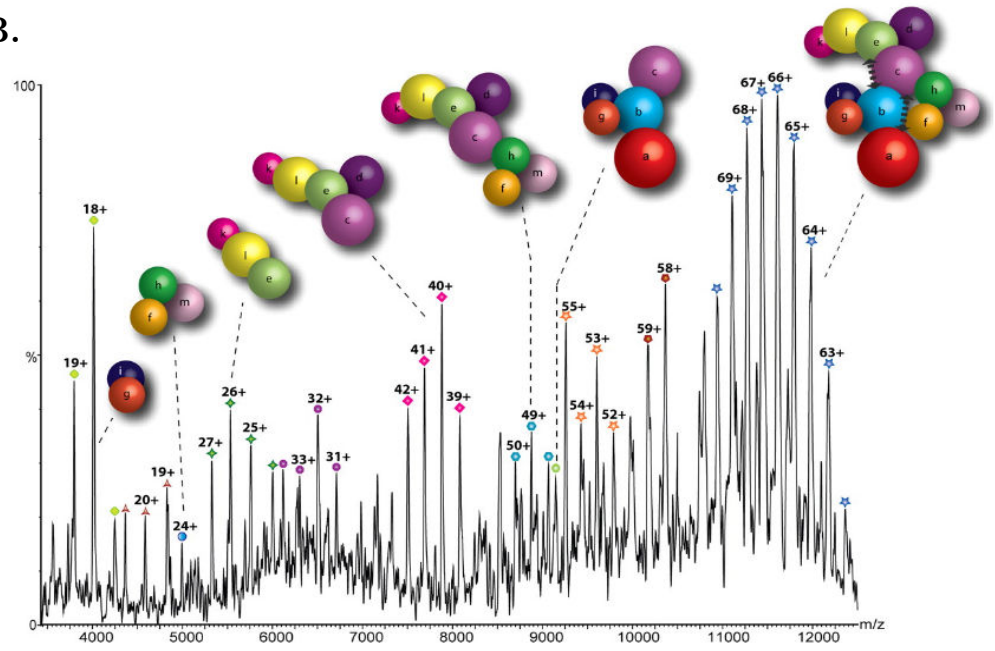


Figure 3.15: **(A) Typical Mass Spectrum of EIF3 complex recorded at ionic strength of 350 mM AmAc. (B) Annotated peaks showing charged states and assignment of complexes from the Mass Spectrum shown in (A).** Figure in (A) (modified) and (B) from [ZSF+08].

3.4.4 Specific pipeline for Assemblies: Top-down approach

Following steps are undertaken to completely characterize the protein assembly [SAR12], [Sha10], [YWZ+12].

Deconvolution of the mass spectrum. A typical spectrum obtained from native mass spectrometry of multiprotein complexes is shown in the Fig. 3.15(A). The spectrum is congested and has various low abundance peaks as well. The aim here is to identify all peaks in the spectrum and determine the charged states. From the charges of the peaks masses of the analyte under scrutiny is determined. To determine the charge states, for all possible sub-complexes (oligomers), their charge states is simulated. It is then matched with that of the original spectrum to annotate the charge distribution for most abundant peaks (Fig. 3.15(B)). Once the charge states are determined the masses are determined by considering two consecutive peaks differing by a charge of 1 unit. For illustration, let the mass of the analyte be M and the charge states being considered are n and $n-1$. Let, $(m/z)_1$ and $(m/z)_2$ are the mass-to-charge values for the above two peaks of charge n and $n-1$ respectively. The two equations can be written as follows:

$$\frac{M+n}{n} = (m/z)_1 \tag{3.6}$$

$$\frac{M+n-1}{n-1} = (m/z)_2 \tag{3.7}$$

From the above two equations, mass of the analyte M can be calculated. The mass of the analyte can be determined from various charge states of the same analyte. The average of all such calculated masses would give more accurate mass of the analyte. Similarly, masses of all the subcomplexes in the spectrum are calculated.

Stoichiometry Determination (SD). From the masses of the subcomplexes, their composition is determined. The problem is to find combination of protein subunits such that their added sum is equal to mass of the sub-complex determined above. This is called stoichiometry determination (SD) problem. For the whole assembly, knowing the mass of the intact assembly and that of each protein subunits, the copy number of each protein subunit is determined such that the added sum is equivalent to the mass of the intact assembly. This problem is discussed in more detail in the chapter on Stoichiometry determination problem. In this manner all the peaks in the spectrum are annotated with the charge states and their respective sub-complexes (Fig. 3.15(B)).

Identification of core and Peripheral Subunits. Tandem MS (MS/MS) analysis is employed to identify the subunits present at the periphery or which are buried in the core of the assembly. The assembly is made to undergo stepwise dissociation by increasing the accelerating voltage. During the flight, the collisions with neutral molecules, e.g., helium, nitrogen or argon, result in the detachment of the protein subunits most exposed, therefore lying on the outer surface. The subsequent analysis of these detached subunits and those retained in the assembly identifies respectively, peripheral and the core subunits.

Connectivity Inference. In a protein macro-molecular assembly, due to 3D spatial constraints, only a subset of all the possible pairwise interactions exists. To infer these existing interaction, the assembly is undergone dissociation in the partial denaturing conditions, through change in ionic strength, pH or adding organic solvents, thereby generating a population of overlapping subcomplexes. These overlapping subcomplexes encode the pairwise interactions existing in the intact assembly. Each of these subcomplexes is assigned to their respective protein subunits using the first two steps of this specific pipeline, e.g. Stoichiometry Determination and Identification of core and peripheral subunits. An appropriate algorithm is then used to infer the pairwise interactions following processing of these overlapping subcomplexes.

Another way of inferring protein-protein interfaces in the complex is through Cross-linking mass spectrometry (CXMS). In this method, solution of protein complex is mixed with a cross linker, e.g. BS^3. Using this linker, lysine residues of the neighboring proteins are linked. A lysine residue has a flexible 6 Åside chain

and BS^3 has a 11.4 Åspacer arm. Therefore, the two Cα atoms must be within 24 Åor so in order for them to be cross-linked. It is then followed by protease digestion yielding highly complicated mixture containing regular, mono-linked, loop-linked and interlinked peptides. Among them the interlinked peptides are the most informative and encoding the information on protein-protein interactions. The mass spectrometry of this mixture thereby produces complex spectra and is subjected to analysis. The inevitable challenges in this approach are that they require special cross-linkers to yield controlled digestion providing some apriori information. Other challenge is to develop faster algorithm to search large databases.

3D model of the assembly. Above connectivity inference provides with the information on topology of the protein subunits within the assembly. The information regarding the shape in 3D can be obtained using Ion-Mobility Mass Spectrometry (IM-MS).

Assembly Dynamics. MS technique holds another distinct advantage to provide with the information on dynamics of protein assemblies whereas, other techniques provide static snapshots. This is due to fast time scale (ms) and ability to simultaneously detect multiple populations of protein assemblies.

3.5 Two Specific Algorithmic Challenges: Mass Decompositions and Connectivity Inference

Here we comment upon two specific algorithmic challenges arising from the data acquisition using mass spectrometry.

3.5.1 Mass Decompositions

Problem statement. For any given molecular system whose building blocks are known, mass measurement can be help us figure out the composition of building blocks. Upon measuring the mass of the intact system and each of its building blocks, the problem reduces down to solve a linear inequality with unknown coefficients representing their respective composition. Mathematically, for an intact system with mass M accurate upto the measured error ε and masses of the p building blocks are respectively, w_1, w_2 .... w_p, then all the possible composition vectors $S = (s_1, \ldots, s_p)$ are solution of the following inequality:

$$| \sum_{i=1,\ldots,p} s_i w_i - M | \leq \varepsilon, \qquad \text{where} \quad \varepsilon, w_i, M \in \mathbb{R}. \tag{3.8}$$

State-of-the-art. The most simple case of the above Eq. (3.8) is when all masses are integers, namely $M, \{w_i\} \in \mathbb{I}$ and $\varepsilon = 0$. Then the inequality becomes a linear equation with unknown coefficients. This resembles money changing problem or coin changing problem. The problem is NP-complete when the intact mass, M and the $\{w_i\}$ vary. In order to solve the above problem, a search tree algorithm was proposed by in which components are added up with over a range of compositions and the solution is found when the added mass equates to the target mass, M. However, the time complexity of this approach is $O(M^{p-1})$ [DF80]. Another way of decomposition is through a dynamic programming algorithm in which a binary table is constructed which works as an oracle to answer whether some mass, M^- could be decomposed into i building blocks, w_1, w_2 .... w_i, where $i \leq p$. All decompositions are then computed using search tree like method in which the tree is explored further only if the oracle approves. The advantage of such a setting is that no sterile leaves are generated and the algorithm is output sensitive. The time complexity to construct the table is $O(p.M)$ and for computing all decompositions is $O(\gamma(M).\frac{1}{w_1}.M)$ where $\gamma(M)$ is the number of solutions. The space complexity for the table is $O(p.M)$ [BL05a]. Another algorithm based on Extended residue table (ERT) was proposed in [BL05a]. The ERT is constructed using the Extended Round Robin's algorithm which could also be used to determine the Frobenius number [BL05b]. Using the ERT table all the decompositions can be determined in $O(p.w_1.\gamma(M))$ time. Running time for table construction is $O(p.w_1)$ and it occupies $O(p.w_1)$ space. The advantage of this algorithm is that it is output sensitive and the running

time is independent of the target mass, M. The measurement of mass is accompanied by error, ε thereby, requiring to solve the interval problem mentioned above in the Eq. (3.8). For the case in which all the masses $M, \{w_i\} \in \mathbb{I}$, the interval problem with integer masses is dealt in the literature with decomposing each mass separately in the interval $[M - \varepsilon, M + \varepsilon]$ [BLM+08]. If the masses $M, \{w_i\} \in \mathbb{R}$, then the above problem in the Eq. (3.8) into an equivalent integer interval problem by multiplying each of the mass by a blowup factor, b, i.e. $\lceil bM \rceil, \{\lceil bw_i \rceil\}$. The bounds of the interval range are corrected using a relative rounding error considerations. There is a need for algorithm to decompose the interval range without having to decompose each mass separately.

3.5.2 Connectivity Inference

Problem statement. Given a macro-molecular assembly whose individual molecules (proteins or nucleic acids) are known, we aim at inferring the connectivity between these molecules. In other words, we are given the vertices of a graph, and we wish to figure out the edges it should have. To constrain the problem, we assume that the composition, in terms of individual molecules, of selected complexes of the assembly is known. Mathematically, this means that the vertex sets of selected *connected subgraphs* of the graph sought are known. To see where this information comes from, recall that a given assembly can be chemically denatured i.e. split into complexes by manipulating the chemical conditions prior to ionization. In extreme conditions, complete denaturation occurs, so that the individual molecules can be identified using MS. In milder conditions, multiple overlapping complexes are generated: once the masses of the proteins are known, the list of proteins in each such complex is determined by solving the aforementioned SD problem [SR07]. As a final comment, it should be noticed that in inferring the connectivity, *smallest-size networks* (i.e. graphs with as few edges as possible) are sought [ADV+07, THS+08]. Indeed, due to volume exclusion constraints, a given protein cannot contact all the remaining ones, so that the minimal connectivity assumption avoids speculating on the exact (unknown) number of contacts.

State-of-the-art. The connectivity inference problem was first addressed in [THS+08] using a two-stage algorithm, called *network inference* (`NI` in the sequel). First, random graphs meeting the connectivity constraint are generated, by incrementally adding edges at random. Second, a genetic algorithm is used to reduce the number of edges, and also boost the diversity of the connectivity. Once the average size of the graphs stabilizes, the pool of graphs is analyzed to spot highly conserved edges.

From the Computer Science point of view, MCI is a network design problem in which one wants to choose a set of edges with minimum cost to connect entities (e.g., routers, antennas, etc.) subject to particular connectivity constraints. Typical examples of such constraints are that the subgraph must be k-connected, possibly with minimum degree or maximum diameter requirements (see [Rag95] for a survey). Such network design problems are generally hard to solve. To the best of our knowledge, the problem of ensuring the connectivity for different subsets of nodes has not been addressed before.

3.6 Thesis overview

We now sketch the contributions presented in chapters 4, 5 and 6.

3.6.1 Stoichiometry determination

To the best of our knowledge, the problem of the determination of the composition of a macromolecular assembly taking into account the errors in the mass measurement has only been touched upon indirectly. In [BLM+08], the interval SD problem is solved by repeatedly calling the exact SD algorithm on each target mass in the interval. In [THS+08], 100 hypothetical complexes from 6 to 14 sub-units have been created with masses in the range 10-50kD, and 100 dimers, trimers or tetramers were generated from these complexes. With an error rate of 1%, it is observed that no tetramer has a unique stoichiometry solution.

In this context, we present two algorithms that inherently address the interval stoichiometry determination problem. First, a constant memory space algorithm (`DIOPHANTINE`), and an output sensitive dynamic programming based algorithm (`DP++`). The performance of these algorithms against previously developed algorithm are made on biological datasets and synthetic datasets. The experiments are run for a range of error values and some inherent properties of these algorithms are unveiled. We observe that large number of solutions exist in the case of biological complexes even for small error values and our algorithms are faster than the previous brute force methodology for the interval case. We also provide results on metabolite datasets which have floating point mass values. We observe that in the previous work it was required to transform the floating point values to integer values using a blow up factor, b and then determine an optimum blow up factor b_{opt} to speed up the calculations. Our algorithm `DIOPHANTINE`spare us of such complicated pre-processing of the input.

3.6.2 Connectivity Inference: the Un-weighted Case

Upon determining the composition of the system by solving the stoichiometry determination problem we move on to the next level and infer the pairwise interactions between subunits of the assembly. We call the problem Minimum Connectivity Inference problem (MCI) which aims at determining the smallest set of contacts solving for the set of oligomers (sub-complexes) obtained from partial denaturation of the assembly. We develop two algorithms; a mixed integer linear programming formulation (`MILP`) and a greedy algorithm. These algorithms are run on the data acquired by mass spectrometry for different biological systems.

We show that the solutions of `MILP` and `Greedy` are more parsimonious than those reported by the algorithm initially developed in biophysics, which are not qualified in terms of optimality (number of contacts used to connect the oligomers). Since MILP outputs a set of optimal solutions, we introduce the notion of *consensus solution*. Using assemblies whose pairwise contacts are known exhaustively, we show an almost perfect agreement between the contacts predicted by our algorithms and the experimentally determined ones, especially for *consensus solutions*.

3.6.3 Connectivity Inference: the Weighted Case

In the earlier case while solving the MCI problem no a priori information on contact is used to find the possible solutions. The native mass spectrometry data could be complimented with various other biophysical experiments such as cross-linking, co-immunoprecipitation, tandem affinity purification etc. They provide additional information on likelihood of a pairwise interaction although with different confidence values depending on the technique used. In this regard, we introduce an extension of MCI problem called *Minimum Weight Connectivity Inference* (MWCI), which involves optimization of functional intermixing on number of contacts and weights on the contacts assigned based on some a priori information.

As discussed above, *consensus solutions* have high precision with respect to the known set of reference contacts. We develop a bootstrap procedure, MILP-W_B that aims to enrich the initial set of contacts in the *consensus solutions*. The idea is as follows; we suppress these initial contacts either by assigning lower weights or forbidding them altogether to hinder their sampling. The problem is then solved for the same set of oligomers but with a smaller pool of contacts which prompts the system to seek alternate connectivity. The new set of contacts in the consensus solutions would enrich the initial set thereby improving the coverage of prospective true contacts. On our test example of yeast exosome, we observe very significant improvement in the performance as compared to the results by the previous *Network inference* algorithm published in [THS$^+$08].

Chapter 4

Stoichiometry Determination Problems

4.1 Introduction

4.1.1 Structural Proteomics and Mass Decompositions

Mass spectrometry: from analytical chemistry to structural proteomics. Mass spectrometry (MS) is an analytical technique allowing to determine the composition of a sample of material into its constituting *characters* (atoms or molecules), based on mass-to-charge (m/z) spectra of ions produced from the sample. Generically, a mass spectrometer consists of three instruments, namely a source creating the ions, a mass filter separating the ions according to their m/z ratio, and a detector which records the charge induced by the passing ions. These different compartments are discussed in further detail in the section 3.2.3. A number of instruments have been designed over the years, two classical ones being the *time-of-flight* (TOF) and the *orbitrap* spectrometers. While the former separate the ions via the measurement of their time of flight (which depends on their mass and charge), the latter resorts to an oscillating electric field so as to selectively focus on specific m/z ratios. To disentangle mixtures containing ions with identical m/z ratio, an iterative protocol known as tandem mass spectrometry can be used. For example, the precursor ions undergo a collision induced dissociation, which consists of removing highly charged peripheral sub-units thanks to gas collisions, so that the product ions can indeed be sorted by m/z ratio. For more detailed description of tandem mass spectrometry and its applications, refer to section 3.2.5. Upon determining the masses of the ions from the m/z spectrum, one is left with a mass decomposition, also known stoichiometry determination (SD) problem), which consists of computing how many copies of each character are needed to account for the target mass.

Mass spectrometry proved instrumental in metabolomics [DAH07]. For example, typical mass range of metabolites in the KEGG database is < 1000 Daltons [1]. Identifying a small molecule consists of finding its elemental composition.The monoisotopic masses of the compounds are in general not sufficient to do so, so that isotope patterns are also used [KF06]. First all the possible organic formulas are determined from the monoisotopic mass and then for each candidate organic formula, its isotopic pattern is simulated and matched with the input pattern to eliminate spurious elemental composition candidates [BLLP09]. In addition to this fragmentation pattern obtained from gas phase fragmentation reactions, e.g. collision induced dissociation (CID) can be exploited. Together with the masses of the product ions and and their relative abundances, fragmentation pattern provides a fingerprint for the precursor ion and can be used to identify ionised molecules [KRCM+12].

More recently, mass spectrometry also proved invaluable to elucidate the stoichiometry of proteins in-

[1] The unified atomic mass unit, also known as Dalton, is approximately equal to the mass of one neutron or one proton.

volved in complexes varying in mass, size, solubility, and bound/unbound states [SR07, BR11]. Typical mass of a protein complex composed of around 10 proteins can weigh several 100 kDa.

Uncertainties in mass determination. Mass analyzers such as orbitraps, which were initially designed to handle small molecules, typically achieve relative mass error of 0.0005% (5 part-per-million or ppm). Interestingly, it has been reported recently that upon modifying the instruments, in particular the ability to completely desolvate the samples [RDD+12], intact macromolecular assemblies with molecular weights of the order 1 MDa were amenable to studies with a precision of 50ppm. Yet, this tour de force has been achieved so far on very few macro-molecular complexes as follows, IgG antibody (146 kDa), bacteriophage HK97 (253 kDa), yeast 20S proteasome (730 kDa), *E. coli* GroEL (801 kDa).

On the other hand, TOF analyzers accommodate a much wider range of samples, and continue to be used for characterization of large macromolecular assemblies [ZCGB13]. Using such instruments, a relative mass error as high as $\sim 1\%$ are typical for hetero-oligomeric complexes of up to 1 MDa [THS+08, MR12]. For such systems, two difficulties are practically faced. First, because the proteins involved in different copies of a complex may have undergone different post-translational modifications [2] [Kel04], there might be a discrepancy between the masses of the sub-units. Second, any mass experimentally determined is generally larger than the theoretical one, due to extra molecules (solvent and electro-spray buffer molecules) sticking to the structure analyzed.

The magnitude of this second source of error actually depends on the analyser and activation energy in the collision cell. Activation in the collision cell can be used to strip off ions and/or buffer adducts from the analyte of interest thus improving spectral quality. For a sufficiently high energy of activation, fragmentation occurs which confirms the mass assignment and also provide structural constraints on the arrangement of subunits [SH14]. We now formalize the SD problems arising due to such errors.

4.1.2 Mass Decompositions: Float Type and Integer Type Problems

Mass decompositions: float type problems. A stoichiometry determination (SD) problem is specified by the individual masses of the p characters $\{w_i\}_{i=1,\ldots,p}$, together with the target mass M. Due to experimental errors, these numbers are floating point numbers. Let the error due to inaccuracy in mass measurement be ε. Then the SD problem is to find all stoichiometry vectors $(s_1, \ldots, s_p)$ such that

$$l \leq \sum_{i=1,\ldots,p} s_i w_i \ \leq u \qquad where,\ l = M - \varepsilon,\ u = M + \varepsilon \tag{4.1}$$

The masses of the atoms constituting a molecule are typically known up to a given precision (e.g. 10^{-9} for *Hydrogen*). This SD problem, in which masses are rational numbers (i.e., a rational number in $\mathbb{Q}$), can be converted into an equivalent integer problem by multiplying all masses by a large enough integer, k (e.g., $\geq 10^9$ for the above mentioned example), s.t. $\lceil kl \rceil = kl$, $\lceil kw_i \rceil = kw_i$ and $\lceil ul \rceil = ul$, which yields:

$$kl \leq \sum_{i=1,\ldots,p} s_i\ kw_i \leq ku. \tag{4.2}$$

However, this transformation leads to very large target masses, which is intractable for many algorithms whose time complexities depend on the target mass. Fortunately, this difficulty can be circumvented using smaller multiplying values called blow-up factors [BLLP09]. Using a blow-up factor $b \in \mathbb{R}$, the float type values are converted into integer ones as $\lceil bw_i \rceil$ and $\lceil bM \rceil$. Because

$$\sum_{i=1,\ldots,p} s_i\ \lceil bw_i \rceil \geq \sum_{i=1,\ldots,p} s_i\ bw_i, \tag{4.3}$$

[2] The chemical modification of a protein after its bio-synthesis by the ribosome.

the upper and lower bounds of Eq. (4.1) must be extended to ensure that all solutions of the original problem are found. Let Δ be the largest relative rounding error defined as,

$$\Delta = \Delta(b) = max\{\Delta_i\}, \quad \text{with } \Delta_i = \Delta_i(b) := \frac{\lceil bw_i \rceil - bw_i}{w_i}, i = 1, 2..., p. \tag{4.4}$$

The SD problem of Eq. (4.1) associated with the blow-up factor b is defined as

$$\lceil bl \rceil \leq \sum_{i=1,\ldots,p} s_i \lceil bw_i \rceil \leq \lfloor bu + \Delta u \rfloor. \tag{4.5}$$

As a result of extending the upper bound certain solutions to the Eq. (4.5) are false positives which need to be filtered out using Eq. (4.1). Also, note that if $b = k$, Eq. (4.4) yields $\Delta(b) = 0$, so that the Eqs. (4.5) and (4.2) are identical.

Choosing a suitable blow-up factor is a delicate problem. On the one hand, using small blow-up factors leads to integer problems having more solutions than their original real-valued problems, i.e., false positive decompositions that need to be removed in a post processing step. On the other hand, increasing the blow-up factor tends to decrease the number of false positive mass decompositions, but it also leads to higher target masses and thus to larger computational times. Therefore, an optimum blow-up factor has to be chosen to address above two issues. Given a max limit $B \in \mathbb{R}$, the following criteria is used to determine a locally optimum blow-up factor [DLMB13]:

$$\text{Find } b \leq B \text{ such that } \quad \frac{\Delta u}{bu} = \frac{\Delta}{b} \quad \text{is minimum.} \tag{4.6}$$

Mass decompositions: integer range problems. Assume, from now on, that the character masses and the target mass are integers. Denoting ε_1 and ε_2 two mass shifts, solving Eq. (4.5) actually requires finding all stoichiometry vectors $(s_1, \ldots, s_p)$ such that

$$M - \varepsilon_1 \leq \sum_{i=1,\ldots,p} s_i w_i \leq M + \varepsilon_2. \tag{4.7}$$

The error free case, namely $\varepsilon_1 = \varepsilon_2 = 0$ is known as the *money change problem* for obvious reasons:

$$\sum_{i=1,\ldots,p} s_i \; w_i = M. \tag{4.8}$$

This problem was first solved using an algorithm exploring all possible stoichiometry vectors in a tree-like fashion [DF80], and later via extensions of dynamic programming [BL05b]. Mathematically and since the masses are integers SD is related to the theory of integer partitions [Com74], to linear diophantine equations [Sma98], and is also coupled to so-called knapsack and subset sum problems [Pis05].

In the sequel, we wish to solve the problem of Eq. (4.7) as a whole, instead of solving one instance of Eq. (4.8) for each mass in the interval.

4.1.3 Contributions

To the best of our knowledge, the question of mass accuracy for the complex composition to be determined has only been touched upon indirectly. In [BLM$^+$08], the interval SD problem is solved by repeatedly calling the exact SD algorithm on each target mass in the interval. In [THS$^+$08], 100 hypothetical complexes from 6 to 14 sub-units have been created with masses in the range 10-50kD, and 100 dimers, trimers or tetramers were generated from these complexes. With an error rate of 1%, it is observed that no tetramer has a unique stoichiometry vector.

In this context, we make two contributions. First, we present a constant memory space algorithm (`DIOPHANTINE`), and an output sensitive dynamic programming based algorithm (`DP++`), both inherently

addressing the interval SD problem. For DIOPHANTINE, we also show that sorting the masses w_i yields a tree of optimal size, an observation which had not been made previously. Second, we present a detailed experimental study on various biological and synthetic datasets, which results in three conclusions. The first observation is that for biological datasets, enumeration does matter even for a moderate (and sometimes even null) noise level. The second finding is that our algorithms DIOPHANTINE and DP++ outperform state-of-the art dynamic programming based approaches by three to four orders of magnitude, for a noise level in the range 0.1% to 1%, which is typically faced in structural proteomics. Not surprisingly, we show that this improvement owes to the ability of our algorithms to avoid redundant calculations which are carried out when calling the exact dynamic programming based algorithm on each target mass in an interval. The last one is that DIOPHANTINE actually exhibits an output sensitive behavior. Thus and interestingly, we show that one of the very first algorithms designed to solve the exact SD problem [DF80] can be modified not only to solve the interval SD problem, but also to outperform advanced strategies based on dynamic programming.

4.2 Theory and Algorithms

In section 4.2.1, we recall the rich background of knapsack and integer partition problems. In section 4.2.2, we present classical counting results, and derive bound on the number of solutions for the SD problem.

4.2.1 Denumerants, Unbounded Knapsack and Subset-sum Problems

In the following, we consider a vector $\mathbf{W} = \{w_1, \ldots, w_p\}$, of positive integers or real numbers. A stoichiometry vector is denoted $\mathbf{S} = \{s_1, \ldots, s_p\}$. The vector is called positive if $s_i > 0$ for all i, and non-negative if $s_i \geq 0$ for all i.

Denumerants. Assume that the vector $\mathbf{W}$ contains integers. In combinatorics, the number of non-negative integer solutions to Eq. (4.7) is known as the *denumerant* $D(M)$ of the target mass. It has been known since Bell [Bel43] that if $\text{lcm}(\mathbf{W})$ denotes the least common multiple of the w_is, then, for each $b \in \{0, \ldots, \text{lcm}(\mathbf{W}) - 1\}$, if $M = m\ \text{lcm}(\mathbf{W}) + b$, the denumerant is a polynomial in m of degree $p-1$, that is:

$$D(m\ \text{lcm}(\mathbf{W}) + b) = \sum_{i=0,\ldots,p-1} c_i m^i. \tag{4.9}$$

More generally, the number of solutions reads from the power series expansion of the generating function $1/\Pi_i(1 - X^{w_i})$, but the difficulty precisely relies in extracting the coefficient of X^n in that expansion. In analytic combinatorics [FS09, Chapter IV], a classical result obtained by singularity analysis states that if $\gcd(\mathbf{W}) = 1$, one asymptotically gets

$$D(M) \sim \frac{M^{p-1}}{(p-1)! w_1 \ldots w_p}. \tag{4.10}$$

Upper and lower bounds on the denumerant have also been obtained based on binomial coefficients solution of certain recurrence relations [Agn02]. However, these bounds are not of real interest for the SD problem for two reasons: first, the bounds concern a particular denumerant rather than the solutions corresponding to an interval, as specified by Eq. (4.7); second, as we shall see with experiments, the bounds returned are not tight.

Frobenius numbers. The asymptotic behavior given by Eq. (4.10) apparently contradicts the existence of (small) masses which cannot be decomposed, this issue being related to the so-called Frobenius number and its generalizations. The Frobenius number $g_0(\mathbf{W})$ is defined as the largest integer which cannot be represented as a non-negative integer combination of the integers in $\mathbf{W}$ [Alf05]. While explicit formula are known up to $M = 3$, only upper and lower bounds are known in general [Alf05]. Computing the Frobenius number is a NP-hard problem [RA96] for which pseudo polynomial time algorithms exist, such as the one by Round-Robin [BL05b]. (The running time of such an algorithm is polynomial with respect to the numerical values of the input data, but exponential in their bit-length.)

A related number is the positive Frobenius number $g_0^+(\mathbf{W})$, namely namely the largest integer which does not admit any positive integer solution. As observed in [BDF+10, Lemma 4], both numbers satisfy:

$$\text{Either } g_0^+(\mathbf{W}) = g_0(\mathbf{W}) = 0, \text{ or } g_0^+(\mathbf{W}) = g_0(\mathbf{W}) + \sum_{i=1,\ldots,p} w_i. \tag{4.11}$$

Unbounded knapsack (UKP) and subset-sum (SSP). Assume that the weights in $\mathbf{W}$ are real numbers coding the weights of p object types, and that each object also has a value $v_i \in \mathbb{R}^+$. Given a target mass M, the unbounded knapsack problem (UKP) consists of finding for each type the integral quantity $s_i \geq 0$, such that the corresponding sum of values is maximum while the corresponding sum of weights does not exceed M. This is formally defined by the following integer programming model:

$$\text{Maximize } \sum_{i=1}^{p} s_i\ v_i, \tag{4.12}$$

$$\text{Under the constraint } \sum_{i=1}^{p} s_i \; w_i \leq M. \tag{4.13}$$

The special case where $w_i = v_i$ holds is known as the subset-sum problem (SSP), and consists in finding the quantities s_i such that the corresponding sum of weights is the closest to M, without exceeding it. Changing Eq. (4.13) into an equality allows subset-sum to solve the exact SD problem of Eq.(4.8).

UKP and SSP: algorithms and complexity. The optimization version of SSP is one of the first problem proved to be NP-Hard [Kar72], but it is also known to be pseudo-polynomial. An example of such pseudo-polynomial time algorithms is well known Bellman recursion [Bel57], which solves UKP in O(M p) time. More recently, an output sensitive algorithm solving SSP has been developed [BL05a]. Denoting w_1 the smallest mass, the algorithm has complexity $O(p\ w_1\ D(M))$, and relies on a data structure of size $O(p\ w_1)$. We shall use our implementation of this algorithm, called DECOMP, as main contender.

Other approaches have been developed to solve UKP, either based on dynamic programming, on branch and bound, or on a combination of both [PYA09]. Interested readers are referred to [PKP04] for detailed review on the different knapsack problems and on the different techniques used to solve them.

4.2.2 UKP and SSP: on the Number of Solutions

Non-negative solutions. The number of solutions $\#\text{UKP}(p, \mathbf{W}, M)$ to an unbounded knapsack problem or of a subset-sum problem that is constrained by Eq. (4.13) was first studied by Beged-Dov [Bd72], and the bounds were later refined by Padberg [Pad71] and Lambe [Lam74].

Denote $\#SD(p, \mathbf{W}, M, \varepsilon)$ the number of solutions to the SD problem defined by the vector $\mathbf{W}$, and by the target mass $M \pm \varepsilon$. The knowledge of this value is of interest in the context of stoichiometry determination, to avoid generating a large number of solutions — which would be impossible to analyze. From the exact number of solutions to an UKP problem, one gets

$$\#SD(p, \mathbf{W}, M, \varepsilon) = \#\text{UKP}(p, \mathbf{W}, M + \varepsilon) - \#\text{UKP}(p, \mathbf{W}, M - \varepsilon - 1). \tag{4.14}$$

However, the previous works [Bd72, Pad71, Lam74] only provide bounds. If # counts an exact number of solutions, denote $\underline{\#}$ and $\overline{\#}$ a lower and an upper bound on #, respectively. Using upper and lower bounds on $\#\text{UKP}(p, \mathbf{W}, M)$, one gets the following lower bound

$$\#SD(p, \mathbf{W}, M, \varepsilon) \geq \underline{\#}\text{UKP}(p, \mathbf{W}, M + \varepsilon) - \overline{\#}\text{UKP}(p, \mathbf{W}, M - \varepsilon - 1), \tag{4.15}$$

together with the following upper bound

$$\#SD(p, \mathbf{W}, M, \varepsilon) \leq \overline{\#}\text{UKP}(p, \mathbf{W}, M + \varepsilon) - \underline{\#}\text{UKP}(p, \mathbf{W}, M - \varepsilon - 1). \tag{4.16}$$

In a nearby vein, a fully polynomial-time approximation scheme (FPTAS) has been derived to estimate the number of solutions of any knapsack problem [GKM$^+$11]. It provides an upper bound that is at most $(1 + \delta)\#$, where δ is an input parameter, and has a time complexity that is polynomial in M and in $1/\delta$. Using the properties of this upper bound, we can easily derive a lower bound and use them with Eqs. (4.15) and (4.16) to bounds SD. In the experiments section, we assess the ability of these strategies to estimate the number of solutions of real stoichiometry determination problems.

Positive solutions. In line with the definition of the shifted Frobenius number of Eq. (4.11), the question of estimating the number of positive solutions is also of interest, in particular for the biological cases where positive stoichiometries are prescribed for selected protein types. To estimate the number of such solutions, it is sufficient to compute the previous bounds with a target mass reduced by the minimal imposed stoichiometries.

4.2.3 Output Sensitive Algorithm

An output sensitive algorithm is an algorithm whose running time depends on the size of output instead or in addition to the size of the input, see e.g. `http://en.wikipedia.org/wiki/Output-sensitive_algorithm`. A simple example of an output sensitive algorithm is *division by subtraction* - for two positive numbers N and D giving remainder R and quotient Q, the run time is O(Q). Another example of an output sensitive algorithm is computing of convex hull of n points in the plane, with k extreme points. The time complexity of this algorithm is $\mathrm{O}(\mathrm{n}\log k)$. Yet another example is the enumeration of all maximal cliques in a graph, see [CK08].

4.3 Solving Float Type Problems via Tree Like Enumeration

The exact case. A pedestrian way to compute the denumerant consists of exhaustively trying all solutions [DF80, MHFH10, MR12]. To solve the interval SD problem of Eq. (4.7) using the same principle, assume that the stoichiometry vectors are built incrementally—say from left to right, and that the stoichiometry vector $\mathbf{S}$ has been computed up to index i (see also Fig. 4.1). Under this assumption, we define the remaining mass to be accounted for by the $p-i$ proteins whose stoichiometry has to be determined by

$$M_0^-[\mathbf{S}] = M, \text{ and } M_i^-[\mathbf{S}] = M - \sum_{j=1,\ldots,i} s_j w_j \text{ for } i = 1,\ldots,p-1,\ M_i^-[\mathbf{S}] \geq 0. \tag{4.17}$$

Then, denoting $a \mid b$ the fact that the integer a divides the integer b, the naive counting strategy for the denumerant consists of computing the following nested sum:

$$\sum_{s_1=0}^{\lfloor M/w_1 \rfloor} \sum_{s_2=0}^{\lfloor M_1^-[\mathbf{S}]/w_2 \rfloor} \cdots \sum_{s_{p-1}=0}^{\lfloor M_{p-2}^-[\mathbf{S}]/w_{p-1} \rfloor} I(w_p : s_1, s_2, s_3 \ldots . s_{p-1}) \tag{4.18}$$

with

$$I(w_p : s_1, s_2, s_3 \ldots . s_{p-1}) = \begin{cases} 1 & if, w_p | M_{p-1}^-[\mathbf{S}] \\ 0 & \text{otherwise} \end{cases} \tag{4.19}$$

The interval case. To solve Eq. (4.7) using the generalized sum of Eq. (4.18), observe that upon reaching the last sigma, the following condition holds:

$$M - \varepsilon - \sum_{j=1,\ldots,p-1} s_j w_j \leq s_p w_p \leq M + \varepsilon - \sum_{j=1,\ldots,p-1} s_j w_j, \tag{4.20}$$

or equivalently

$$\lceil \frac{M - \varepsilon - \sum_{j=1,\ldots,p-1} s_j w_j}{w_p} \rceil \leq s_p \leq \lfloor \frac{M + \varepsilon - \sum_{j=1,\ldots,p-1} s_j w_j}{w_p} \rfloor \tag{4.21}$$

Therefore, denoting $\#(\cdot)$ the length of an integer interval, counting the number of stoichiometry vectors solving the SD problem is done by

$$\sum_{s_1=0}^{\lfloor (M+\varepsilon)/w_1 \rfloor} \sum_{s_2=0}^{\lfloor (M+\varepsilon)_1^-[\mathbf{S}]/w_2 \rfloor} \cdots \sum_{s_{p-1}=0}^{\lfloor (M+\varepsilon)_{p-2}^-[\mathbf{S}]/w_{p-1} \rfloor} I'(w_p : s_1, s_2, s_3 \ldots . s_{p-1}) \tag{4.22}$$

with

$$I'(w_p : s_1, s_2, s_3 \ldots . s_{p-1}) = \#_{s_p}(\lceil (M-\varepsilon)_{p-1}^-[\mathbf{S}]/w_p \rceil \leq s_p \leq \lfloor (M+\varepsilon)_{p-1}^-[\mathbf{S}]/w_p \rfloor) \tag{4.23}$$

Note that when $\varepsilon = 0$, the equations (4.22-4.23) and (4.18-4.19) coincide.

Remark 1. *For the sake of conciseness, given a stoichiometry vector* $\mathbf{S}$*, let* $\sum_{\mathbf{S}} = \sum_{s_i \in \mathbf{S}} s_i w_i$*. Also, let the* remaining mass *associated to* $\mathbf{S}$ *be defined as* $m = M + \varepsilon - \sum_{\mathbf{S}}$*. One has:*

$$M - \varepsilon \leq \sum_{\mathbf{S}} \leq M + \varepsilon \Leftrightarrow 0 \leq \sum_{\mathbf{S}} -M + \varepsilon \leq 2\varepsilon \Leftrightarrow 0 \leq m \leq 2\varepsilon. \tag{4.24}$$

These equivalent conditions characterize a valid stoichiometry vector.

Algorithm `DIOPHANTINE`**.** To turn Eq. (4.22-4.23) into an algorithm enumerating the solutions, observe that the $i+1$th Σ consists of finding the stoichiometry vector indices in the range $i+1, \ldots, d$, which accounts for the remaining mass $(M+\varepsilon)_i^-[\mathbf{S}]$. Algorithm 1 presents `DIOPHANTINE`, the corresponding recursive procedure. The recursion tree explored by this algorithm is presented on Fig. 4.1. Its size, defined as its number of edges, represents the cost of the execution. A leaf is called fertile if it yields a solution, and sterile otherwise. Likewise, an edge is called fertile if it leads to at least one fertile leaf, and sterile otherwise. We note in passing that when the masses are sorted in descending order, one case of output-sensitivity is easily detected. Indeed, if

$$w_p = w_{min} \leq 2 \times \varepsilon, \tag{4.25}$$

all leaves are fertile—the length of the interval defined in Eq. (4.20) is 2ε so that is contains at least one solution for s_p.

Regarding the memory footprint, because the stoichiometry vector $\mathbf{S}$ is passed by reference, a unique vector is used along the whole recursion tree, and we have:

Observation. 1. *Algorithm* `DIOPHANTINE`*takes* $\Theta(p)$ *storage.*

Remark 2 (On positive versus non-negative solutions.)**.** *As seen from the sigmas of Eq. (4.22), algorithm* `DIOPHANTINE`*generates all non-negative solutions. If minimal stoichiometries for the proteins are imposed, compliant solutions can be generated by starting the summations at these stoichiometries—in particular, staring at 1 rather than 0 yields positive solutions. Such solutions can also be generated by subtracting the mass corresponding to these constraints from the target mass, and seeking non-negative solutions.*

On the sortedness of weights. The cost of a particular enumeration problem is provided by the number of edges of the recursion tree explored by `DIOPHANTINE`, which actually depends on the sortedness of the vector $\mathbf{W}$ of weights:

Observation. 2. *The size of the recursion tree of algorithm* `DIOPHANTINE`*is minimized when the vector of weights is sorted in descending order.*

Proof. The proof is divided into three steps. First, we prove that swapping two consecutive weights w_i and w_{i+1} only changes the number of node at depth i. Second, that swapping w_i and w_{i+1} results in a smaller number of node at depth i only if after reordering, $w_i \geq w_{i+1}$. Third, that iteratively applying this results leads to the optimal weight reordering $w_1 \geq w_2 \geq \cdots \geq w_p$.

Point 1. Consider the upper bound of Eq. (4.20). Without reordering, a node of the exploration tree at depth $k < p$ is a solution to $\sum_{i=1}^{k} s_i \times w_i \leq M + \varepsilon$, and the number of such node is equal to the corresponding number of such solutions (which we denote here by $\#_k$). For depth p, $\#_p$ is the number of solution as specified by Eq. (4.23). Thus, the number of tree node is $\#_1 + \#_2 + \cdots + \#_p$. Let now swap w_i and w_{i+1}. The numbers of nodes at depth 1 to $i-1$ do not depend on w_i nor on $w_i + 1$, and thus do not change. $\#_{i'}$, the new number of nodes at depth i is now the number of solutions to $\sum_{j=1}^{i-1}(s_j \times w_j) + s_{i+1} \times w_{i+1} \leq M + \varepsilon$, which is different from $\#_i$. The number of nodes at depth $i+1$ is the number of solutions to $\sum_{j=1}^{i-1}(s_j \times w_j) + s_{i+1} \times w_{i+1} + s_i \times w_i \leq M + \varepsilon$, which is the same as $\#_{i+1}$, and the same holds for larger depths.

Algorithm 1 Algorithm `DIOPHANTINE`**.** The algorithm stores the stoichiometry vectors in a set *Sol*. The recursive function takes four arguments : the stoichiometry vector $\mathbf{S}$ under construction, passed by reference as indicated by the ampersand & (C++ convention); the remaining mass $m = M_{i-1}^{-}[\mathbf{S}]$ of Eq. (4.17); the error threshold ε; and the index $i \in [1, \ldots, p]$ of the protein type to be processed by the current recursive call. The weights are sorted in the decreasing order. The initial call is `DIOPHANTINE`$(\mathbf{S_0}, M + \varepsilon, \varepsilon, 1)$, where $\mathbf{S_0}$ is a stoichiometry vector whose p entries are set to 0, M is the target mass, and ε is the allowed error. The sub-routine `keep_value` checks whether the stoichiometry j of the ith protein is admissible—as one may impose a lower and/or upper bound on the stoichiometry of that type.

```
DIOPHANTINE(Stoi_vector & S, unsigned m, unsigned ε, unsigned i)
// Stop the recursion if we are on the last protein type
if i == p then
  q_min = ⌈(m − 2ε)/w_p⌉ // p-th type: min stoichiometry
  q_max = ⌊m/w_p⌋        // p-th type: max stoichiometry
  for j = q_min to q_max do
    if keep_value(p, j) then
      S[p] = j
      Insert S into Sol

  // Recurse to set the remaining stoichiometries
else
  quotient = ⌊m/w_i⌋ // i-th type: max stoichiometry
  for j = 0 to quotient do
    if keep_value(i, j) then
      S[i] = j //Set the i-th stoichiometry
      DIOPHANTINE(S, m − (j w_i), ε, i + 1) // Recursion
```

Point 2. For any branch defined by a sub-solution $\mathbf{S} = (s_1, ..., s_{i-1})$, the number of nodes at depth i is $\lfloor (M + \varepsilon)_{i-1}^{-}[\mathbf{S}]/w_i \rfloor + 1$. This number is smaller for larger values of w_i, and is thus minimum if we reorder w_i and w_{i+1} so that $w_i \geq w_{i+1}$

Point 3. The optimal weights reordering is then by decreasing order of weights (i.e. when $w_1 \geq w_2 \geq \cdots \geq w_p$). This is proved easily by remarking that using any other weights reordering suppose that there exist a i for which $w_i < w_{i+1}$, implying that the number of nodes in the corresponding exploration trees can be decreased by swapping protein types i and $i + 1$. □

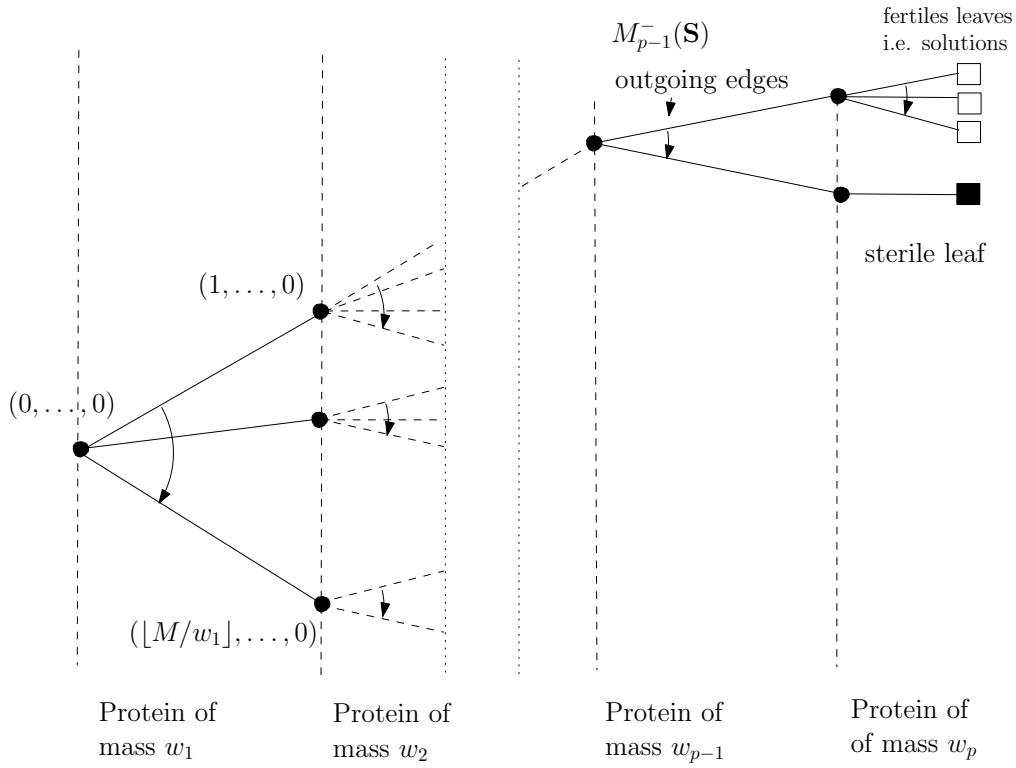


Figure 4.1: **The recursion tree explored by algorithm** `DIOPHANTINE`**.** A leaf is called fertile if Eq. (4.23) has at least one solution, and sterile otherwise. Similarly, any edge of the tree is termed fertile if at least one fertile leaf is found downstream the tree, and sterile otherwise. The size of the recursion tree is defined as its number of edges.

4.4 Solving Integer Type Problems with Dynamic Programming

The classical solution to the money changing problem based on dynamic programming is well known [BL05a]. It uses a binary table stating whether the target mass is decomposable using a subset of proteins, this table being used by the backtracking algorithm reporting all solutions. In the sequel, we show that a slight modification of the construction of the binary table based on Eq. (4.21) helps in accommodating the interval case. The enumeration algorithm using this table shall be denoted `DP++`, and is called as DP++$(M + \varepsilon, p)$.

Assume that the weights w_i are sorted by increasing value. Along the course of the backtracking, denote m the mass remaining from $M + \varepsilon$ once a set of protein instances have been used. That is, given the stoichiometry vector $\mathbf{S}$, one has remaining mass,

$$m = M + \varepsilon - \sum_{s_i \in \mathbf{S}} s_i w_i \tag{4.26}$$

We wish to build a binary table whose semantics is the following:

$$B[i, m] = 1 \quad \Leftrightarrow \quad \exists \text{ a mass } x \text{ in the interval } [max(0, m - 2\epsilon), \ldots, m] \\ \text{such that } x \text{ is decomposable over } \{w_1, w_2, ...w_i\}.$$

Denote $\#s_1$ the number of non-negative integer values satisfying $\lceil (m - 2\varepsilon)/w_1 \rceil \leq s_1 \leq \lfloor m/w_1 \rfloor$. For

$i = 1, \ldots, M + \varepsilon$ and $i = 1, \ldots, p$, the binary table $B[i, m]$ as follows:

$$\text{For } i = 1 : B[1, m] = 1 \text{ if } \#s_1 \geq 0, \text{ and 0 otherwise} \tag{4.27}$$

$$\text{For } i > 1 : B[i, m] = \begin{cases} B[i-1, m] \text{ if } m < w_i \\ B[i-1, m] \vee B[i, m - w_i] \text{ otherwise.} \end{cases} \tag{4.28}$$

`DP++`consists of backtracking as usual using this binary table. `DP++`is also output sensitive, a property inherited from the original algorithm. The number of backtracking steps for each solution being equal to $(M + \varepsilon)/w_1$ in the worst case, if $D(M, \varepsilon)$ denotes the total number of solutions, the enumeration runs in time $O((M + \varepsilon)\ D(M, \varepsilon)/w_1)$.

4.5 Material and Methods

In this section, we introduce the datasets used for our integer type problems typically coming from structural proteomics, and for float type problems typically related to small molecules.

4.5.1 Comparison: Methodology

For integer type problems, the focus is on positive solutions. In structural proteomics, the protein types involved in a given protein complex can generally be determined by mass spectrometry upon dismantling the complex under stringent conditions, so that the individual subunits are known (whence a minimal stoichiometry of one).

For float type problems, the focus is on non-negative solutions.

Practically, the experiments were conducted on a DELL PRECISION T7400 computer equipped with 8Go of RAM, running Fedora Core 14. For all programs, a cut-time of three hours was set, in order to handle the case of enumeration problems yielding an astronomical number of solutions. For such cases, we estimated the number of solutions using the FPTAS strategy of [GKM$^+$11]. In fact, whenever the 3 hours time limit was hit, we obtained a lower bound on the number of solutions by taking the maximum value of the number of solutions computed by `DIOPHANTINE`upon hitting the cut-time, and the lower bound computed from the FPTAS.

In analyzing running times, for algorithms `DECOMP`and `DP++`, the total running was split into pre-processing (binary table building), and post-processing (enumerating the solutions), e.g. $t^{\text{Tot}}_{\texttt{DECOMP}} = t^{\text{Pre}}_{\texttt{DECOMP}} + t^{\text{Post}}_{\texttt{DECOMP}}$.

4.5.2 Datasets for Integer Type Problems

Real Datasets

In the following, we briefly present `Data-Bio` dataset, which consists of 10 biological complexes, with masses span the range 321,274 – 5,276,467 Da (Table 4.1). This table lists experimental molecular weights, except for NPC (Y-ring and single spoke), where theoretical molecular weights were computed from the known stoichiometries. The reader is referred to the supplemental section SI-4.12.1 for more details.

Yeast 19S Proteasome lid. Proteasomes are protein assemblies involved in the elimination of damaged or misfolded proteins, and the degradation of short-lived regulatory proteins. The most common form of proteasome is the 26S, which involves two filtering caps (the 19S), each cap involving a peripheral lid, composed of 9 distinct protein types each with unit stoichiometry [STA$^+$06].

COP9 Signalosome. The COP9 Signalosome is a multi-functional complex primarily involved in ubiquitin mediated proteolysis linked to diverse cellular activities such as signal transduction, cell cycle progression and transcriptional regulation. It is composed of 8 protein protein types each with unit stoichiometry.

Eukaryotic Translation factor EIF3. Eukaryotic initiation factors (eIF) are proteins involved in the initiation phase of the eukaryotic translation. They form a complex with the 40S ribosomal subunit, initiating the ribosomal scanning of mRNA. Among them, EIF3 consists of 13 different protein types each with unit stoichiometry.

Yeast Exosome. The exosome is a 3'- 5' exonuclease complex involved in RNA processing and degradation. The yeast exosome is composed of 10 different protein types with unit stoichiometry [HDT$^+$06].

Rotary ATPases. Rotary ATPases are membrane associated molecular machines involved in energy conversion by coupling ATP hydrolysis (or synthesis) with proton (or Na^+) translocation across biological membranes. Practically, we investigate the intact TtATPase, and four sub-complexes of EhATPase.

Yeast Nuclear Pore Complex (NPC). The NPC, which is the largest protein assembly known to date in the eukaryotic cell [WR10, DH08], is a protein assembly anchored in the nuclear envelope, regulating the nucleo-cytoplasmic transport. It is composed of $\sim$ 30 distinct proteins types each present in multiple copies. It has eight-fold radial symmetry and consists of eight spokes, each containing 57 protein instances. One particular complex, the Y-complex, involving 7 protein types with unit stoichiometry, is present in two sets of 8 copies. Each such set presumably forms one ring (the Y-ring) involving 56 protein instances [FMPS$^+$12, DDC12].

Artificial Datasets

Prime numbers instances: the `Data-Prime10` dataset. The use of prime number weights prevents relative multiplicity of individual weights, and therefore avoids outright hardness of the instance stemming from so-called spanner sets [Pis05]. To generate the kth synthetic complex, with $k \in 1, \ldots, 10$, p prime numbers were chosen randomly from the list of prime numbers from 7,000 to 70,000, in accordance with the weights (in Daltons) of individual proteins involved in the biological complexes of `Data-Bio`. Moreover, 20 instances for each kth weight vector(W_k) were generated by using 20 target masses uniformly spaced in the range

$$[\sum w_i,\ 2g_0^+(\mathbf{W_k})]. \tag{4.29}$$

These 200 instances are referred to as `Data-Prime10-i-j`.

Random biological instances: the `Data-Pseudo-Bio10` dataset. To generate synthetic complexes using the masses of real proteins, we replaced the set of prime numbers by the masses of $\sim$ 6700 YEAST proteins retrieved from the non-redundant UNIPROT protein database, see `http://www.uniprot.org/help/about`. The theoretical molecular weight of proteins were calculated by adding up the weight of constituent amino acids in the proteins. (To be consistent with `Data-Prime10`, only masses beyond 7 kDa were retained.) As above, 10 vectors of masses were picked, with 20 target masses in the interval of Eq. (4.29), yielding 200 instances denoted `Data-Pseudo-Bio10-i-j`.

Error levels. Three error levels typical in structural proteomics were used, namely 0%, 0.1% and 1% — the value ε of Eq. (4.7) being set to the error level times the target mass for the following three datasets – `Data-Bio`, `Data-Pseudo-Bio10` and `Data-Prime10`.

4.5.3 Dataset for Float Type Problems

Real Datasets

This dataset contains 653 compounds downloaded from the MassBank database [HAK$^+$10] at `http://www.massbank.jp/`, accession numbers PR100001 to PR101056 (1719 spectras) curated by Dr. Masanori Arita at PSC, RIKEN. The ion type of all compounds is $[C+H]^+$, i.e. singly charged ions. The monoisotopic mass

of each compound is considered with an accuracy of 20 ppm, so $\epsilon = 0.00002M$, where M is the monoisotopic mass of the compound. The blow up factor, b is taken to be 10^5. The alphabets of decomposition are chosen to be organic elements, Carbon (C: 12.000000 Da), Hydrogen (H: 1.007825032 Da), Oxygen (O: 15.99491462 Da), Nitrogen (N: 14.00307401 Da), Phosphorus(P: 30.9737615 Da), Sulphur(S: 31.9720707 Da), Chlorine (Cl: 34.968853), Bromine (Br: 78.918337) and Iodine (I: 126.904473).

Among the above 1719 spectras, there are 459 unique monoisotopic masses. Target masses are decomposed into two alphabets $CHONSP$ and $CHONSPClBrI$ resulting in two datasets respectively, `Data-MassBank6`and `Data-MassBank9`.

We also decompose 62 unique monoisotopic masses from the EAWAG Dataset (`Data-Eawag9`), 98 unique monoisotopic masses from the HILL Dataset (`Data-Hill9`) and 90 unique monoisotopic masses from the ORBITRAP Dataset (`Data-Orbitrap9`). All of them are decomposed to $CHONSPClBrI$.

Artificial Datasets: the `Data-SynMetab6`and `Data-SynMetab9`datasets

We designed instances for synthetic metabolites. These synthetic metabolites were generated for the alphabets $CHONSP$ (size 6) and $CHONSPClBrI$ (size 9). In each case, 100 instances were generated, with target masses uniformly spaced in the range as per the Eq. (4.29).

Blow-up factor, b

To transform the SD problem with floating point numbers into an integer SD problem, a blow-up factor is to be used as in the equation 4.1. The following two blow-up factors are chosen randomly – 30,000 and 100,000 and using equation 4.6 the corresponding optimum blow-up factors (b_{opt}) are respectively, 29105.2 and 96687.4. We also provide the results for the blow up factor, $b = 5963.33$ recommended in [DLMB13]. As far as algorithm `DIOPHANTINE`is concerned, the SD problem with floating point numbers is solved by multiplying the SD inequality by a large enough integer, $k = 10^9$ as in equation 4.2.

4.6 Results: Integer Type Problems

In this section, we report observations on the number of solutions of integer interval type problems — rather than on algorithms to solve them.

4.6.1 Biological Examples: Enumeration Matters Even at Null Noise Level

The formula recalled in Eq. (4.10) shows an asymptotic polynomial growth of the number of solutions, yet, are situations with multiple solutions commonplace in biology? Table 4.1 answers this question positively, with multiple solutions even at null noise level in some cases. It is to be noted that masses of the intact assemblies measured are corrupted by chemical noise and that is why looking for the solutions at null noise level does not have any solution for most of the protein assemblies as can be seen in the column for #sol at 0% noise level of the Table 4.1. The case of the NPC is interesting. The case of one full spoke shows that the stoichiometry determination problem in this case is ill-posed, with an astronomical number of solutions. The same holds for the NPC-Y-ring system, which consists of 8 copies of the Y-complex, since a total of 788 solutions are obtained.

The last column of this table also shows that a large value M/g_0^+ is a good hint at a large number of solutions. Note that, for the value of the total mass beyond g_0^+, there is certainly a decomposition with positive stoichiometries. Hence, greater the ratio M/g_0^+, more likely is the rise in number of decompositions. From the Table 4.1, One full spoke and Y ring of NPC have M/g_0^+ respectively, 1.66, 2.02 reflecting on the large number of solutions for both.

Also, a solution sketch for positive solutions, as defined in section 4.6.3, is presented on Fig. 4.2 for the system EhATPase-sub-4. While on this simple example every node of the Hasse diagram corresponds to a single solution, other example yield more complex (and cluttered) Hasse diagrams, with numerous solutions per node. Interestingly, we also observed that for non-negative solutions and even a 0.1% noise level, all

solution sketches of biological complexes pretty much involve all possible tuples (supplemental Figs. SI-4.23 and SI-4.24). That is, the Hasse diagram essentially contains all $\sum_{k=1}^{p} \binom{p}{k}$ nodes which are defined by Eq. 4.31.

Remark 3. *In structural proteomics, the protein types involved in a given protein complex can generally be determined by mass spectrometry upon dismantling the complex into its individual sub-units, thanks to denaturing conditions. This explains why Table 4.1 is about positive solutions—rather than non-negative solutions. But all the remaining experiments have been conducted with non negative solutions.*

4.6.2 Counting Solutions and Convergence to the Denumerant

Rationale. As noticed in section 4.2, counting solutions ahead of computing them is of interest to avoid generating a large number of (useless) solutions. We compared the accuracy of the estimations provided in [Pad71, Bd72, Lam74] against those of the FPTAS algorithm of [GKM+11].

One comment is also in order about *hard instances*, i.e. instances such that `DIOPHANTINE`does not terminate within a prescribed time limit—three hours in this work. According to [Pis05], two parameters lead to hard UKP and SSP instances: the magnitude of weights, and the linear redundancy between the weights. Intuitively, if a given weight is a linear combination of other weights, additional solutions can be generated. However, beyond the Frobenius number and in the asymptotic regime, the number of solutions of SSP is given by Eq. 4.10. Therefore, instead of using a target mass proportional to the sum of masses, as in [Pis05], we systematically explored ranges of target masses expressed in units of Frobenius number.

Results: counting solutions. The biological examples just discussed show that estimating the number of solutions is important to detect ill posed problems.

Table 4.2 compares various strategies to estimate the number of solutions to a SD problem. First, we plugged the upper and lower bounds provided by [Pad71], [Bd72], and [Lam74] into Eqs. (4.15-4.16). For each biological system, we retained the best bounds obtained—those yielding the tighter interval (Table 4.2, columns 2-4). Second, we also computed estimates using the FPTAS of [GKM+11] (Table 4.2, columns 5-7).

On the one hand, the combinatorial bounds are not of real interest, since at least two orders of magnitude of difference between $\underline{\#}\mathrm{UKP}(p, \mathbf{W}, M)$ and $\overline{\#}\mathrm{UKP}(p, \mathbf{W}, M)$, yielding a null lower bound. On the other hand, the bounds from the FPTAS are much tighter.

Results: convergence towards the denumerant. Due to the rapid rise of the number of solutions, an interesting problem is the speed of the convergence of the number of solutions of a SD problem to the denumerant.

To study this convergence, a toy dataset called `Data-Quad4` consisting of the quadruplet $\{10, 15, 32, 48\}$ from [SS11] was used. A total of 100 instances uniformly distributed in the range $[\sum w_i, 41g_0]$ was used, the jth instance being denoted `Data-Quad4` $- j$. We also studied this convergence for a real biological system of Yeast Exosome complex. A total of 40 instances having target masses uniformly distributed in the range $[\sum w_i, 20g_0]$. The results on this system and on a real biological system are presented on Figs. 4.3 and 4.4, and show a rapid convergence to the denumerant. The ratio of denumerant to the number of non-negative solutions grows to 96% at $40g_0$ starting from 29%, in case of `Data-Quad4` whereas, for the real biological instance, this ratio grows to 70% at $10g_0$ from its initial value of 12%. The Yeast Exosome instances having target mass greater than $11g_0$ did not terminate within the time limit of 3h. Note that to compare with Denumerant, non-negative solutions are plotted.

4.6.3 Solution Sketches to Represent a Solution Set

Because an interval SD problem may admit a large solution set $\mathcal{D}$, the question arises to represent all such solutions in a compact way. For a solution $\mathbf{S} \in \mathcal{D}$, we define the stripped solution $\underline{\mathbf{S}}$ by

$$\underline{\mathbf{S}} = \{\ldots, \underline{s_i}, \ldots\} \text{ with } \underline{s_i} = s_i - m_i, \text{ and } m_i = \min_{\mathbf{S} \in \mathcal{D}} s_i. \tag{4.30}$$

Note that the stoichiometries m_i define the *background* of $\mathcal{D}$. In addition, we define the skeleton $\mathbf{B}$ of a solution $\mathbf{S}$ by

$$\mathbf{B} = \{\dots, b_i, \dots\} \text{ with } b_i = 1 \text{ if } \underline{s_i} > 0, \text{ and 0 otherwise.} \tag{4.31}$$

The set of ones in a skeleton vector identify the protein types $\mathbf{W_B}$ involved in a stripped solution. Since inclusion among such sets defines a partial order, we construct the following Hasse diagram, called the *sketch* of the solution set $\mathcal{D}$:

- A node corresponds to a set $\mathbf{W_B}$. Each such node is associated the solutions of $\mathcal{D}$ whose skeleton defines $\mathbf{W_B}$. The multiplicity μ of the node is the number of such solutions.

- An edge links the nodes $\mathbf{W_B}$ and $\mathbf{W_D}$ iff $\mathbf{W_B} \subset \mathbf{W_D}$, and there does not exist a node $\mathbf{W_C}$ such that $\mathbf{W_B} \subset \mathbf{W_C} \subset \mathbf{W_D}$.

Note that stripping a solution set $\mathcal{D}$ is mandatory for positive solutions of an interval SD problem — or the sketch would involve a single node corresponding to all protein types.

Assembly	#sol, 0% noise	#sol, 0.1% noise	#sol, 1% noise	M (Da)	g_0^+	$\#sol(1\%\,noise)/2\varepsilon$	M/g_0^+
COP9	0	1	1	321,274 ± 35	961,855	3.12×10^{-4}	0.33
Y-19S-lid	0	0	1	376,151 ± 369	921,712	1.87×10^{-4}	0.41
EhATPase-sub-5	0	4	39	387,356 ± 230	682,901	5.03×10^{-3}	0.57
Y-exosome	0	13	149	402,708 ± 68	649,185	1.85×10^{-2}	0.62
EhATPase-sub-4	0	20	190	424,441 ± 148	682,901	2.38×10^{-2}	0.62
EhATPase-sub-3	0	74	700	461,674 ± 324	682,901	8.10×10^{-2}	0.68
EhATPase-sub-2	0	224	2,213	500,178 ± 294	682,901	2.32×10^{-1}	0.73
TtATPase	21	24,487	246,242	659,202 ± 131	607,304	18.7	1.08
EIF3	0	0	1	797,999 ± 180	1,257,629	8.05×10^{-5}	0.63
NPC-Y-ring	788	6,900,664	69,042,257	4,603,280	2,282,543	750	2.02
NPC-1-spoke	$[1.72 \times 10^{16}, 1.73 \times 10^{16}]$	$[1.72 \times 10^{16}, 4.77 \times 10^{16}]$	$[2.71 \times 10^{17}, 3.44 \times 10^{17}]$	5,276,467	3,169,210	$\geq 2.57 \times 10^{12}$	1.66

Table 4.1: **Biological instances can lead to multiple positive solutions, even with null noise level.** For the NPC-1-spoke, lower and upper bounds were obtained using UKP-FPTAS of [GKM+11] with an error factor of $\delta = 0.1$.

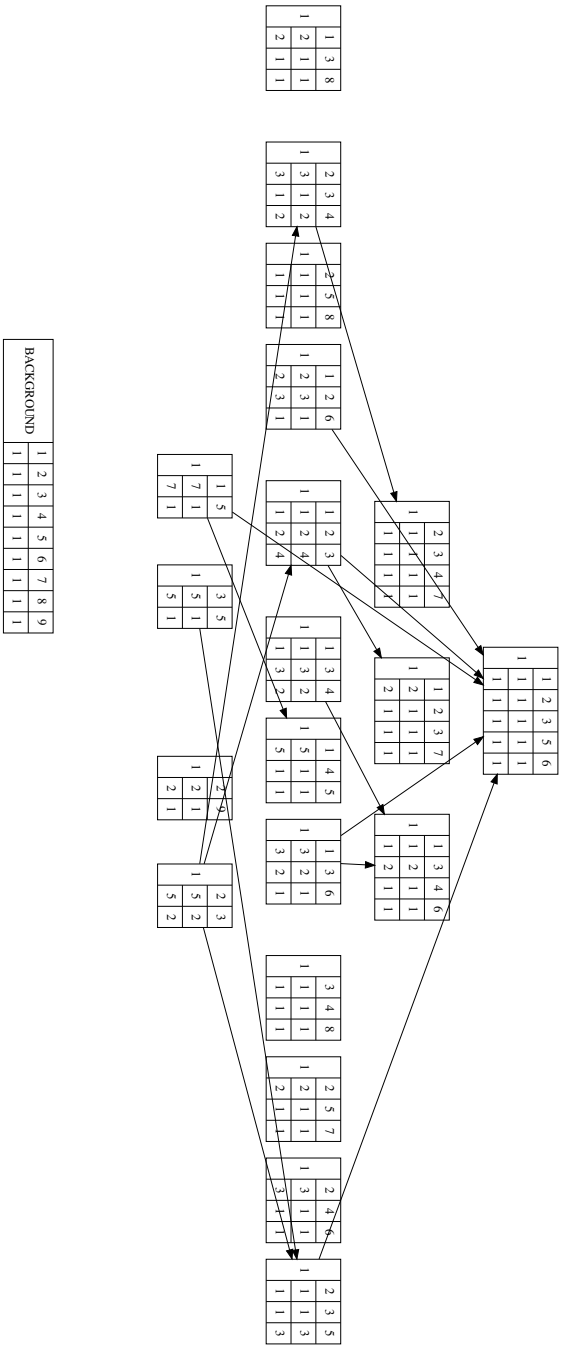


Figure 4.2: **The 20 solutions for EhATPase-sub-4 at 0.1% noise level, see also Table 4.1** The Hasse diagram presents the sketch of the solution set, as defined in section 4.6.3, while the background presents the min stoichiometry of each type present in any solution. Each node of the Hasse diagram reads as follows: the left hand side presents the number of solutions. Each row on the right hand side reads as follows: (top) list of protein types (middle, bottom) min and max stoichiometries of all solutions using these types. In the block for *Background information*, first row corresponds to the index of protein types and the second row corresponds to the minimum stoichiometry of these protein types across all the positive solutions. Note in particular that an arrow represents the inclusion between the protein types of two nodes.

Complex	# solutions	$M \pm 1\%$		$M \pm 1\%$	
		$\underline{\#}SD$	$\overline{\#}SD$	$\underline{\#}SD$	$\overline{\#}SD$
COP9	1	0	8	1	1
Y19-lid	1	0	10	1	1
Y-exosome	149	0	486	101	183
EIF3	1	0	14	1	1
NPC-1-Spoke	Unknown	0	1.06×10^{22}	2.71×10^{17}	3.44×10^{17}

Table 4.2: **Estimating the number of positive solutions for five biological systems.** Columns 2 to 4: upper and lower bounds obtained by plugging the upper and lower bounds of [Pad71], [Bd72], and [Lam74] into Eqs. (4.15-4.16). Columns 5 to 7: upper and lower bounds on the number of positive solutions obtained using the FPTAS of [GKM$^+$11], with $\delta = 0.1$.

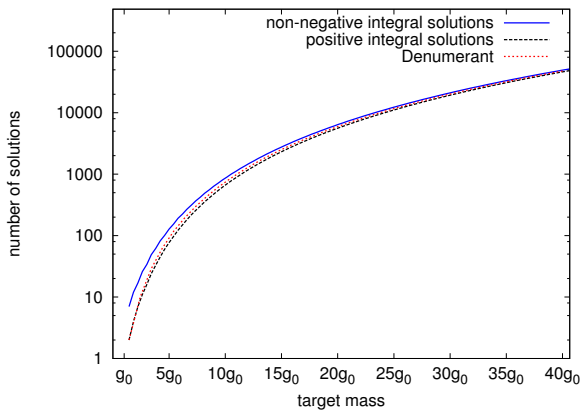


Figure 4.3: **Number of non-negative solutions vs. asymptototic behavior of the denumerant (Eq. (4.10)) for the quadruplet $\mathbf{W} = \{10, 15, 32, 48\}$.** The target mass is expressed in units of the Frobenius number. *Note that Y-axis is drawn with a logarithmic scale.*

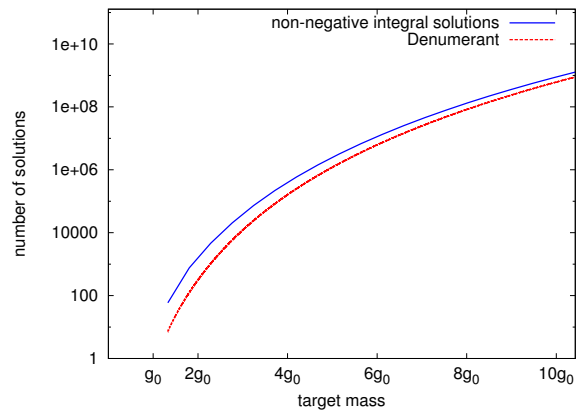


Figure 4.4: **Number of non-negative solutions vs. asymptototic behavior of the denumerant (Eq. (4.10)) for Yeast Exosome.** The target mass is expressed in units of the Frobenius number. Instances with a target mass beyond $11g_0$ did not terminate within the time limit. *Note that Y-axis is drawn with a logarithmic scale.*

4.7 Results: Algorithm DIOPHANTINE– versus DECOMP

4.7.1 Float Type Problems

Decompositions and their relevance

The algorithm DECOMPgenerates high number of false positives even with locally optimum blow up factors, b = 29105.2 and 96687.4, respectively (data not shown). The invalid decompositions are then filtered out using Eq. (4.7) after which the solution set yielded by DECOMPbecomes independent of the blow-up factor and is identical to that yielded by DIOPHANTINEwith $b = 10^9$. For the dataset, Data-Eawag9there are 13/62 non-decomposable instances by either of the algorithms.

Running times. The alphabet for the four metabolic datasets are identical, therefore, the ERT table needs to be constructed once as it only depends on minimum character mass and alphabet size and not on the target mass. Therefore, for a batch run with the algorithm DECOMP, the total run time $t^{\text{Tot}}_{\texttt{DECOMP}}$ has three components, namely (i) the time to construct the table once, (ii) the calculation of optimal blow-up factor, and (iii) the time to enumerate all solutions.

The algorithm DIOPHANTINEoutperforms DECOMPby more than an order of magnitude for batch run time on Data-MassBank9. When the ordering of alphabet masses is reversed running time of DIOPHANTINEincreased by two orders of magnitude (data not shown). Therefore, the naive search tree algorithm which does not take ordering into account performs poorly. (Table 4.3).

The iterative versions of DECOMPand DP++(DP++_ITER_RANGE) were proposed in [DLMB13] along with improved memory requirement for binary table construction used for DP++. From the Table 4.4 it can be seen that DIOPHANTINEis 4 folds faster than DECOMP(with $b = 5963.33$).

In addition, the running time of all the algorithms, other than DIOPHANTINE, clearly depends upon the choice of locally optimal blow-up factor and it is therefore required to test a series of blow-up factors to find the one yielding the lowest running time (Tables 4.3, 4.4). Also, owing to the dependence on the set of character masses, a different set would therefore require to find another blow-up factor yielding the best running time making it less convenient. The algorithm DIOPHANTINEhowever, has several advantages over DECOMPand DP++_ITER_RANGE. Firstly, there is no need for any memory requirement. Secondly, it does not require to find the best locally optimal blow-up factor. One can just multiply with an appropriate power of 10 depending on the maximum precision of the mass measurement as explained in section 4.1.1 on Mass Decomposition for Float problems.

Algorithm	min	median	max	Σ
MassBank				
DIOPHANTINE, $b = 10^9$	~ 0	~ 0	1.22	8.9
DECOMP, $b = 29105.2$	~ 0	0.01	16.53	117.00
DECOMP, $b = 96687.4$	~ 0	0.02	23.65	175.79
Eawag				
DIOPHANTINE, $b = 10^9$	~ 0	~ 0	0.05	0.25
DECOMP, $b = 29105.2$	~ 0	0.01	0.42	3.75
DECOMP, $b = 96687.4$	~ 0	0.15	0.51	6.47
Hill				
DIOPHANTINE, $b = 10^9$	~ 0	~ 0	0.08	1.13
DECOMP, $b = 29105.2$	~ 0	0.04	0.82	14.21
DECOMP, $b = 96687.4$	~ 0	0.04	0.99	17.53
Orbitrap				
DIOPHANTINE, $b = 10^9$	~ 0	~ 0	9.7	20.29
DECOMP, $b = 29105.2$	~ 0	0.04	183.41	352.40
DECOMP, $b = 96687.4$	~ 0	0.04	192.40	378.42

Table 4.3: **Min, median, max and cumulative running times (in seconds) for the algorithms on the entire datasets** `Data-MassBank9`**,** `Data-Eawag9`**,** `Data-Hill9`**,** `Data-Orbitrap9`**.**. The cumulative time is the sum of individual running time of all the instances. Note that the resolution of the timer used is 10 ms. The blow up factors $b = 29105.2$ and $b = 96687.4$ correspond to the optimized values for $B = 30000$ and $B = 100000$ (see Eq. 4.6).

Algorithm	Blow-Up	Orbitrap	MassBank	Eawag	Hill
DIOPHANTINE	10^9	20.15	8.7	0.23	1.18
DECOMP	629.074	258.49	70.95	1.36	6.93
	5963.33	82.42	31.90	1.08	4.23
DP++	629.074	22.10	17.62	1.96	4.37
	5963.33	40.04	138.98	18.55	40.40

Table 4.4: **Running time (in seconds) on a batch of** `Data-MassBank9`**,** `Data-Eawag9`**,** `Data-Hill9`**,** `Data-Orbitrap9`**as a function of the blowup factor** b. The blow up factor, $b = 5963.33$ is recommended in [DLMB13] and $b = 629.074$ correspond to the optimized values of blow up factors for $B = 1000$ (see Eq. 4.6). Note that the resolution of the timer used is 10 ms.

4.7.2 Integer Type Problems

Decompositions and their relevance

The importance of enumerating solutions for integer type problems has already been discussed in section 4.6. Therefore, we focus on performances in the sequel.

Running times

While DIOPHANTINEcompleted all instances within the imparted time (3 h), regardless of the noise level, on a per-dataset basis, DECOMPfinished on 9/10, 109/200, 96/200 cases at 0.1% noise level, and 8/10, 59/200, 54/200 cases at 1% noise level, where the series of numbers corresponds to the three following datasets in that order – `Data-Bio`, `Data-Pseudo-Bio10` and `Data-Prime10`.

The median ratio triplet, $t^{\text{Tot}}_{\texttt{DIO}}/t^{\text{Tot}}_{\texttt{DECOMP}}$ corresponding to three noise levels of 0%, 0.1% and 1% noise level is $(0.775,\ 7.91 \times 10^{-4},\ 9.14 \times 10^{-5})$. Note that the median is computed considering all the instances together from three datasets. Furthermore, at null noise level there is no clear winner among `DIOPHANTINE`and `DECOMP`, although median ratio being 0.775. At 0.1% and 1% noise level, `DECOMP`is not competitive anymore, since it is outperformed by three to four orders of magnitude as reflected in the median ratio values and in the plots (Fig. 4.5).

In order to compute running time over the whole batch of instances from the three aforementioned datasets, since `DECOMP`does not terminate within 3h time limit, we run the instances with 12h time limit. We observe that on a per-dataset basis, `DECOMP`terminated on 9/10, 109/200, 96/200 instances at 0.1% noise-level, i.e. the numbers do not change with increased time limit. However, at 1% noise level, `DECOMP`terminated for 9/10, 100/200, 90/200 instances. The min, median, max triplet along with the batch run time is given in the tables 4.5, 4.6 , respectively for the noise levels of 0.1% and 1%. While comparing the performance of `DIOPHANTINE`with that of `DECOMP`, one observes from median and batch run time values that for the noise-level of 0.1%, `DIOPHANTINE`is atleast 3 orders of magnitude faster than `DECOMP`and for the noise level of 1%, the performance is atleast 4 orders of magnitude faster.

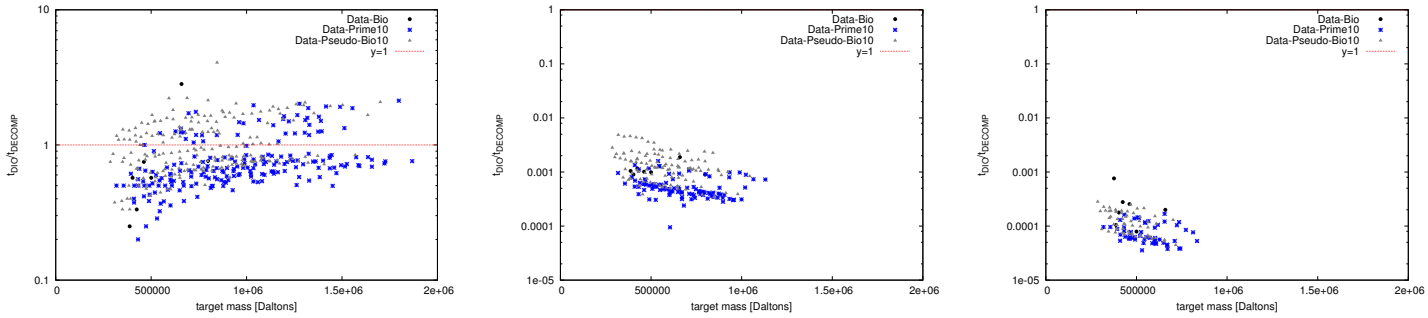


Figure 4.5: `DIOPHANTINE`**vs.** `DECOMP`**: running time as a function of the target mass at different noise levels.** The 3 columns respectively correspond to the noise levels 0%, 0.1%, and 1%. *Note that Y-axes for the figures in the right column are drawn with a logarithmic scale.*

Algorithm	min	median	max	Σ
		Data-Bio		
DIOPHANTINE	~ 0	0.03	1.22	1.87
DECOMP	0.31	0.5m	22.65m	29.73m
DP++	0.19	0.30	1.92	5.54
		Data-Pseudo-Bio10		
DIOPHANTINE	0.01	5.46	257.99	5895.47
DECOMP	9.08	95.23m	$>$ 720m	$>$ 68983.51m
DP++	0.20	15.57	689.27	16684.97
		Data-Prime10		
DIOPHANTINE	0.01	5.55	248.49	6178.18
DECOMP	16.4	$>$ 720m	$>$ 720m	$>$ 78521.63m
DP++	0.22	15.70	798.42	18265.38

Table 4.5: **Min, median, max and cumulative running times (in seconds, unless otherwise stated) for the algorithms on the entire datasets** `Data-Bio`, `Data-Pseudo-Bio10`, `Data-Prime10` **with 0.1% noise level**. The cumulative time is the sum of individual running time of all the instances. Note that the time limit imparted for each instance is 12h i.e. 720m.

Algorithm	min	median	max	Σ
		Data-Bio		
DIOPHANTINE	~ 0	0.03	1.32	1.99
DECOMP	2.52	3.64m	388.95m	454.53m
DP++	0.21	1.74	15.70	43.83
		Data-Pseudo-Bio10		
DIOPHANTINE	0.02	6.065	265.46	6565.42
DECOMP	0.97m	666.85m	$>$ 720m	$>$ 85483.80m
DP++	0.19	17.965	797.59	19062.25
		Data-Prime10		
DIOPHANTINE	0.01	6.18	256.11	7202.25
DECOMP	1.7m	$>$ 720m	$>$ 720m	$>$ 91747.8m
DP++	0.25	19.615	752.80	21715.74

Table 4.6: **Min, median, max and cumulative running times (in seconds, unless otherwise stated) for the algorithms on the entire datasets** `Data-Bio`, `Data-Pseudo-Bio10`, `Data-Prime10` **with 1% noise-level**. The cumulative time is the sum of individual running time of all the instances. Note that the time limit imparted for each instance is 12h i.e. 720m.

4.8 Results: Algorithm DP++– versus DECOMP

4.8.1 Float Type Problems

Decompositions and their relevance

As mentioned previously, both the algorithms DP++(DP++_ITER_RANGE) and DECOMPfor a given blow-up factor, b generate high number of false positives for float type problems. These invalid decompositions or false positives are eliminated using the Eq. 4.7 .

Running times

When the blow up factor, $b = 29105.2$ is used we observe that the algorithm DECOMP outperforms DP++ on the datasets Data-MassBank6. For the datasets Data-MassBank9and Data-SynMetab6, the median value for running time and total batch run time for DP++is greater although the maximum value is lower as compared to that of DECOMP(Tables 4.7, 4.8). It can be seen that as the target mass increases the ratio monotonically approaches 1 (Fig. 4.6).

Now, with the recommeded blow up factor $b = 5963.33$ in [DLMB13], among the four datasets, Data-MassBank9, Data-Eawag9, Data-Hill9, Data-Orbitrap9, the performance of DP++_ITER_RANGEvaries. The DP++_ITER_RANGEis outperformed by DECOMPon other three datasets (Table 4.4) and performs better than DECOMPonly on Data-MassBank9. For the same datasets, if the blow up factor used is $b = 629.074$ (locally optimized value with $B = 1000$), we see that DP++_ITER_RANGEoutperforms DECOMPon the three out of four of the above datasets and on Data-Eawag9it has comparable performance (Table 4.4).

We also examined the incidence of alphabet size on the performance of these two algorithms, DP++_ITER_RANGEand DECOMP, since both are output sensitive. The blow up factor used is $b = 29105.2$. It can be seen that as the alphabet size is increased from "six" to "nine", the increase in total batch running times respectively for $MassBank$ and $Synthetic$ datasets for DECOMP are 13.97 times and 118.77 times, whereas for DP++_ITER_RANGE are 1.12 times and 2.63 times. Also, DP++_ITER_RANGEoutperforms DECOMPby the factor of ~ 32 on the dataset Data-SynMetab9for the total batch running time (Tables 4.7, 4.8). The running time ratios $t^{\text{Tot}}_{\texttt{DECOMP}}/t^{\text{Tot}}_{\texttt{DP++}}$ goes above 1 for 69% of the instances for the alphabet $CHONSPClBrI$ from 19% with alphabet $CHONSP$ (bottom row of the Fig. 4.6).

Time complexity analysis: DP++versus DECOMP

The performance of DP++_ITER_RANGEw.r.t DECOMPis attempted to explain by the ratio of asymptotic complexities,

$$\frac{t^{\text{Tot}}_{\texttt{DP++}}}{t^{\text{Tot}}_{\texttt{DECOMP}}} \sim \frac{O(M\ p\ +\ \frac{M}{w_{min}}\ D(M,0))}{O(w_{min}\ p\ +\ w_{min}\ p\ D(M,0))} \tag{4.32}$$

On simplifying and ignoring the constants, we get,

$$\begin{aligned} &= \frac{M\ p\ (1\ +\ \frac{D(M,0)}{p\ w_{min}})}{p\ w_{min}\ (1\ +\ D(M,0))} \\ &= \frac{M}{p\ w_{min}^2}(1\ +\ \frac{p\ w_{min} - 1}{1\ +\ D(M,0)}) \\ &= \frac{M}{p\ w_{min}^2}(\frac{p\ w_{min}\ +\ D(M,0)}{1\ +\ D(M,0)}) \end{aligned}$$

For instances, s.t., $p\ w_{min} \gg D(M,0) \geq 0$,

$$= \frac{M}{w_{min}}(\frac{1}{1\ +\ D(M,0)}) \tag{4.33}$$

	Data-MassBank6				Data-MassBank9			
	min	median	max	Σ	min	median	max	Σ
DECOMP	~ 0	0.01	0.62	8.37	~ 0	0.01	16.53	117.00
DP++	0.14	0.99	4.19	639.86	0.16	1.14	5.66	714.14

Table 4.7: **Min, median, max and cumulative running times (in seconds) for the algorithms DECOMPand DP++on the entire datasets Data-MassBank6and Data-MassBank9**. The cumulative time is the sum of individual running time of all the instances. The value of locally optimized blow-up factor used is 29105.2. Note that the resolution of the timer used is 10 ms.

	Data-SynMetab6				Data-SynMetab9			
	min	median	max	Σ	min	median	max	Σ
DECOMP	~ 0	0.71	19.63	340.18	0.05	56.77	2708.40	40402.84
DP++	0.55	4.24	8.00	471.91	3.17	8.77	40.78	1240.78

Table 4.8: **Min, median, max and cumulative running times (in seconds) for the algorithms DECOMPand DP++on the entire datasets Data-SynMetab6and Data-SynMetab9**. The cumulative time is the sum of individual running time of all the instances. The value of locally optimized blow-up factor used is 29105.2. Note that the resolution of the timer used is 10 ms.

The excellent correlations between $\frac{O_{\text{DP++}}}{O_{\text{DECOMP}}}$ and $\frac{M}{w_{min}}(\frac{1}{1\ +\ D(M,0)})$ in the Fig. 4.13 for all the four datasets suggest that the governing equation is Eq. 4.33, the assumption being $p\ w_{min} \gg D(M,0) \geq 0$. It should be noted that this assumption can be arrived at by putting $t^{\text{Pre}}_{\texttt{DP++}} >> t^{\text{Post}}_{\texttt{DP++}}$. Therefore, in all these datasets, post processing of DP++is substantially lower than that of pre-processing time. Also, in the Eq. 4.33, the ratio is dependent upon #solutions. Since, beyond the frobenius number of the system each target mass would have a decomposition. The datasets Data-SynMetab6and Data-SynMetab9ensured this regime and the number of solutions would then asymptotically reach the denumerant Eq. 4.9, which in turn is dependent on the alphabet size p. On running the experiments we see that #solutions rose for Data-SynMetab9and thus outperforming the DECOMP(Table 4.8, Fig. 4.6).

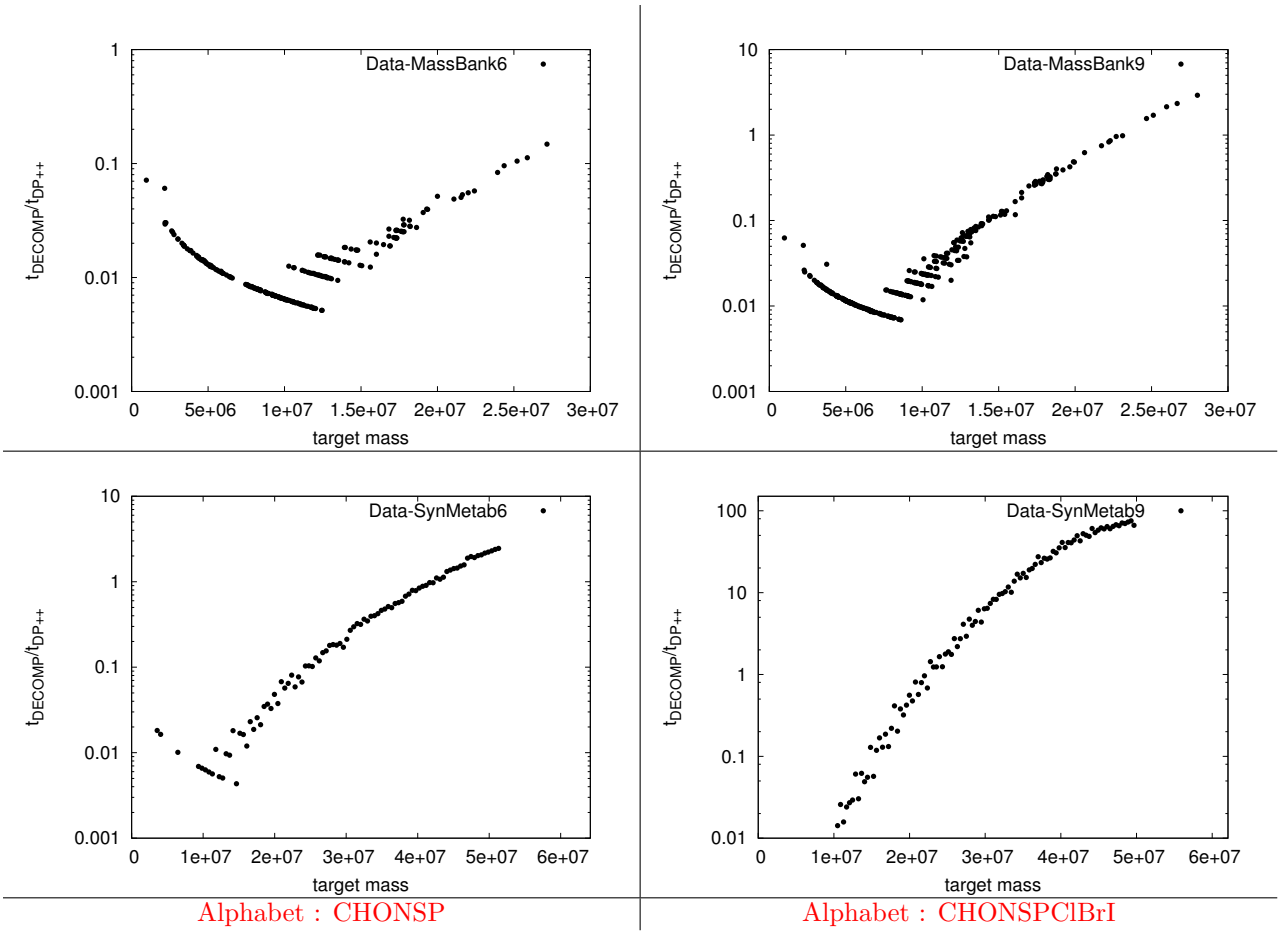


Figure 4.6: **Performances of DP++versus DECOMPfor two alphabet sizes, at noise level is 20ppm.** Top row: MassBank Dataset Bottom row: Synthetic Metabolites Dataset

4.8.2 Integer Type Problems

Decompositions and their relevance

The wider the interval $[M-\varepsilon,\ldots,M+\varepsilon]$, the more challenging the instances for DECOMP, an issue precisely analyzed later in this section.

Running times

The algorithm DP++terminated for all the instances within the imparted time irrespective of the noise level, whereas, DECOMPterminated for fewer instances as mentioned in the subsection on **Integer Type Problems** of Sec. 4.7.

The median ratio triplet, $t^{\text{Tot}}_{\texttt{DP++}}/t^{\text{Tot}}_{\texttt{DECOMP}}$ corresponding to three noise levels of 0%, 0.1% and 1% noise level is $(0.117,\ 3.08\times10^{-3},\ 5.09\times10^{-4})$. Note that the median is computed considering all the instances together from three datasets.

Furthermore, at null noise level, DP++starts to outperform DECOMPas the target mass increases. At, 0.1% and 1% noise level, DECOMPis outperformed by more than 3 orders of magnitude (Fig. 4.7, Tables. 4.5, 4.6).

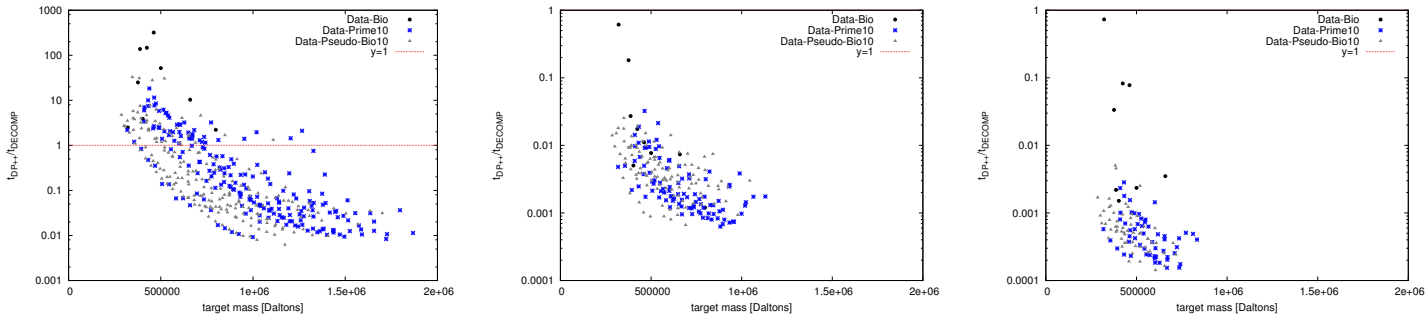


Figure 4.7: DP++vs. DECOMP: **running time as a function of the target mass at different noise levels.** The 3 columns respectively correspond to the noise levels 0%, 0.1%, and 1%. *Note that Y-axes for the figures in the right column are drawn with a logarithmic scale.*

Miscellaneous: Computation tree analysis for DP++and DECOMP.

Rationale. The larger the noise level, the better the performances of DP++w.r.t. DECOMP. Also, algorithm DP++is output sensitive for the interval SD problem, while algorithm DECOMPis output sensitive for the exact SD problem but possibly not for the interval SD problem. We wish to relate these two facts, using two ingredients:

- First, the decomposition of the running time into pre and post-processing, namely $t^{\text{Tot}}_{\texttt{DP++}} = t^{\text{Pre}}_{\texttt{DP++}} + t^{\text{Post}}_{\texttt{DP++}}$, and likewise for DECOMP.
- Second, the number of nodes explored by DP++and DECOMP. For DP++, this number is the number of nodes in the backtracking procedure, as explained in section 4.4. For DECOMP, this number is equal to the sum of the number of explored nodes for target masses in the range $[M - \varepsilon, M + \varepsilon]$.

Results: for algorithm DP++, the post-processing time $t^{\textbf{Post}}_{\texttt{DP++}}$ converges to total runtime $t^{\textbf{Tot}}_{\texttt{DP++}}$ when the number of solutions increases. We first analyze the relative importance of the pre-processing and post-processing times for both algorithms. As shown by the scatter plots (Fig.4.15), the post-processing step always dominates for DECOMP, while for DP++ it becomes dominant as the number of solutions increases. For the latter algorithm, the linear trend between $t^{\text{Post}}_{\texttt{DP++}}/t^{\text{Pre}}_{\texttt{DP++}}$ and the number of nodes but also the number of solutions is confirmed by correlation coefficients beyond 0.9, at both noise levels 0.1% and 1% (Table 4.11, and supplemental Figs. SI-4.16, SI-4.17). Interestingly, the correlation between the ratio of running times $t^{\text{Post}}_{\texttt{DP++}}/t^{\text{Pre}}_{\texttt{DP++}}$ and the target mass is poor, with coefficients of 0.32 and 0.11 at 0.1% and 1% noise level respectively.

Results: the decreasing performance of DECOMPowes to a redundant post-processing. Since algorithm DP++is output sensitive, we take it as a yardstick and postulate that that if the number of nodes explored by DECOMPincurs an x-fold increase with respect to the number of nodes explored by DP++, so should the running-times. Phrased differently, we compute the Pearson correlation coefficient

$$\text{Corr}\left(\frac{t^{\text{Post}}_{\texttt{DECOMP}}}{t^{\text{Post}}_{\texttt{DP++}}}, \frac{\#nodes_{\text{DECOMP}}}{\#nodes_{\text{DP++}}}\right). \tag{4.34}$$

As seen from Table 4.12, this first correlation is strong, with values of 0.75 and 0.79 at noise levels of 0.1% and 1% respectively. On removing two outliers at each of the noise level, the coefficients improve to 0.86 and 0.93, respectively (Supplemental Fig. SI-4.20). The linear relationship between the two terms of the correlation defined by Eq. (4.34), together with the convergence of $t^{\text{Post}}_{\texttt{DECOMP}}$ to $t^{\text{Tot}}_{\texttt{DECOMP}}$, and the convergence of $t^{\text{Post}}_{\texttt{DP++}}$ to $t^{\text{Tot}}_{\texttt{DP++}}$ establish that the increased total running time $t^{\text{Tot}}_{\texttt{DECOMP}}$ linearly depends on the number of nodes visited by DECOMP. In particular, taking as reference $\#nodes_{\text{DP++}}$ since algorithm DP++ is output

sensitive, the ratio $\#nodes_{\text{DECOMP}}/\#nodes_{\text{DP++}}$ measures the redundant work of DECOMP corresponding to the successive calls to solve the individual SD problems.

Since the previous line of argument holds asymptotically, namely when $t^{\text{Post}}_{\texttt{DP++}}$ converges to $t^{\text{Tot}}_{\texttt{DP++}}$, we also investigate directly the Pearson correlation coefficient

$$\text{Corr}(\frac{t^{\text{Tot}}_{\texttt{DECOMP}}}{t^{\text{Tot}}_{\texttt{DP++}}}, \frac{\#nodes_{\text{DECOMP}}}{\#nodes_{\text{DP++}}}). \tag{4.35}$$

With values of 0.48 and 0.35 at noise levels 0.1% and 1%, this correlation is weak. The problem in computing the coefficient of Eq. (4.35) with $N = 203$ [3] and with $N = 112$ [4] instances respectively at 0.1% and 1% noise levels is that the overall correlation is spoiled by the instances for which the pre-processing time still dominates in a run of DP++. To measure the incidence of the convergence of $t^{\text{Post}}_{\texttt{DP++}}$ to $t^{\text{Tot}}_{\texttt{DP++}}$ when the number of solution increases, we therefore resort to a sequence of correlation coefficients, the set of instances used to investigate the correlation corresponding to the cases where the ratio $t^{\text{Post}}_{\texttt{DP++}}/t^{\text{Pre}}_{\texttt{DP++}}$ is larger than some threshold. More precisely, given a set of N instances, we compute N correlation coefficients as follows:

- Based on the running times of algorithm `DP++` we compute

$$\alpha_i = t^{\text{Post}}_{\texttt{DP++}}/t^{\text{Pre}}_{\texttt{DP++}} \tag{4.36}$$

  for each instance, and sort the N instances by increasing α_i value.

- For each index $i = 1, \ldots, N-1$, let

$$S_{>i}: \text{ the set of instances such that } \alpha_j > \alpha_i. \tag{4.37}$$

  We compute the Pearson correlation coefficient $\text{Corr}(\frac{t^{\text{Tot}}_{\texttt{DECOMP}}}{t^{\text{Tot}}_{\texttt{DP++}}}, \frac{\#nodes_{\text{DECOMP}}}{\#nodes_{\text{DP++}}})$ on the set $S_{>i}$.

The plot of these coefficients is presented on Fig. 4.18. If one omits the last 10 coefficients — which are inherently unstable since they involve less than 10 points, the values obtained for the coefficient of Eq. (4.35) now reach 0.9. Two points are noticeable. First, at 0.1% noise level, the correlation rises rapidly as a function of the α_i value. As scatter plots show (supplemental Fig. SI-4.25), the decreasing section for the trailing 30 instances or so owes to the paucity of the dataset. Second, at 1% noise level, the correlation rises monotonically over the range of α_i values explored. This indicates that the increase in number of nodes of `DECOMP`translates directly on the running time $t^{\text{Tot}}_{\texttt{DECOMP}}$, which is expected since when α_i increases, the post-processing time which depends on the number of nodes converges to the total running time. Again, the ratio $\#nodes_{\text{DECOMP}}/\#nodes_{\text{DP++}}$ directly measures the redundancy of the work carried out by algorithm `DECOMP`with respect to the output sensitive algorithm `DP++`, this redundancy having a linear incidence on the running time.

Discussion. These observations are actually consistent with the complexities of the pre-processing steps and the number of nodes generated. Indeed, the pre-processing step in the case of `DECOMP`is independent of the target mass of the instance ($O(p\ w_{min})$), while in case of `DP++`it is linearly dependent on the target mass ($O((M+\epsilon)p)$).

As for the number of nodes generated during the post-processing step, the degraded performances of `DECOMP`for the interval SD problem is actually expected. Indeed, while a given stoichiometry vector is generated exactly once for any interval SD problem (the solutions to two different target masses are different), redundant calculations are possibly carried out by different runs of `DECOMP`for consecutive target masses. Ideally, one would measure the redundancy of calculations carried out to generate two different solutions is precisely measured by the longest common sub-sequence between the stoichiometry vectors of these two solutions. More plainly, we use the number of steps required by the backtracking procedure.

[3] number of instances for which both the algorithms terminated within 3h with 0.1% noise level. Note that `DP++`terminated for all 410 instances.

[4] number of instances for which both the algorithms terminated within 3h with 1% noise level. Note that `DP++`terminated for all 410 instances.

4.9 Results: Algorithm DIOPHANTINEversus DP++

4.9.1 Float Type Problems

Decompositions and their relevance

For a given blow-up factor b, the algorithm DP++(DP++_ITER_RANGE) also generates high number of false positives that are filtered out using Eq. (4.1), whereas, no such false positives are generated by the algorithm DIOPHANTINE.

Running times

The algorithm DIOPHANTINEhas median run time which is comparable to the resolution of the timer, 10 ms. The total batch run time taken by DP++_ITER_RANGEto compute the Data-MassBank9is more than 20 folds and 80 folds respectively, for $b = 29105.2$ and $b = 96687.4$ (Table 4.9).

	min	median	max	Σ
DIOPHANTINE, $b = 10^9$	~ 0	~ 0	4.16	29.24
DP++, $b = 29105.2$	0.16	1.14	5.66	714.14
DP++, $b = 96687.4$	0.5	3.78	16.86	2414.07

Table 4.9: **Min, median, max and cumulative running times (in seconds) for the algorithms on the entire dataset** Data-MassBank9. The cumulative time is the sum of individual running time of all the instances. Note that the resolution of the timer used is 10 ms.

When other blow up factors, $b = 629.074, 5963.33$ are examined, we see that DP++_ITER_RANGEis outperformed by DIOPHANTINEon all the four real datasets of metabolites, Data-MassBank9, Data-Eawag9, Data-Hill9, Data-Orbitrap9, although relatively with a low multiplicative factor. The algorithm DIOPHANTINEis atmost ~ 3.7 times faster than DP++_ITER_RANGEamong above four datasets. Also, it is noted that the choice of blow up factor $b = 629.074$ over $b = 5963.33$ improves the performance of DP++_ITER_RANGEby a factor varying between 2 and 10 (Table 4.4).

4.9.2 Integer Type Problems

Decompositions and their relevance

Running times

Both the algorithms terminated for all the instances within the imparted time (3h). The median ratio triplet, $t^{\text{Tot}}_{\texttt{DP++}}/t^{\text{Tot}}_{\texttt{DIO}}$ corresponding to three noise levels of 0%, 0.1% and 1% noise level is (0.126, 2.89, 3.05) for the three datasets combined together – Data-Bio, Data-Pseudo-Bio10 and Data-Prime10 (Fig. 4.8). Note that the median is computed considering all the instances together from three datasets. Furthermore, at null noise level, DP++starts to outperform DIOPHANTINEwith increase in the target mass. At 0.1% and 1% noise level, DIOPHANTINEovertakes DP++by a factor of ~ 3 (Fig. 4.8, Tables. 4.5, 4.6). Also, it can be seen that the ratio tends to reach an asymptotic value as the error level increases from 0.1% to 1%, hinting at an output sensitive behavior of DIOPHANTINE— recall that DP++is output sensitive (Supplemental Sec. 4.13).

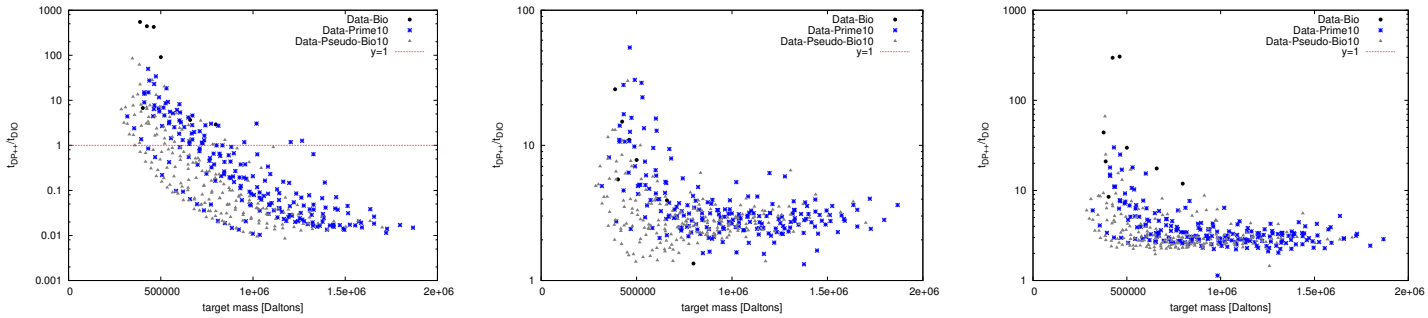


Figure 4.8: DP++vs. DIOPHANTINE: **running time as a function of the target mass at different noise levels.** The 3 columns respectively correspond to the noise levels 0%, 0.1%, and 1%. *Note that Y-axes for the figures in the right column are drawn with a logarithmic scale.*

4.10 Conclusion and Outlook

Mass spectrometry is playing an increasing role to investigate large macro-molecular assemblies, thus complementing more classical techniques such as X ray crystallography or nuclear magnetic resonance, which are better suited for smaller complexes. In this context, the stoichiometry determination (SD) problem, which consists of determining how many copies of each sub-unit are required to account for the observed mass, is the first one to be addressed, before investigating the geometry of the contacts between these sub-units. In this work, we develop a constant memory space enumeration algorithm (DIOPHANTINE), and an output sensitive dynamic programming based algorithm (DP++_ITER_RANGE), which are shown to outperform state-of-the-art SD algorithms by several (three to four) orders of magnitude. These two algorithms exhibit comparable performances for typical noise levels in the range 0.1% to 1%, which is remarkable since DIOPHANTINEdoes not use any pre-processing and has constant memory footprint. Both algorithms performed to satisfaction on all the biological cases processed, which are the most challenging ones currently investigated in structural biology. It is noticeable that enumeration does matter, since systems with numerous solutions are observed at a moderate (sometimes even null) noise level. Also, the coupling between DIOPHANTINE(or DP++_ITER_RANGE) and the fully polynomial-time approximation scheme (FPTAS) of [GKM$^+$11] provides a powerful way to get a lower bound on the number of solutions, and thus to identify ill-posed SD problems—which admit a too large number of solutions.

We have also examined the performance DIOPHANTINEand DP++_ITER_RANGEon small metabolite datasets against previously published algorithm, DECOMP. We observe that DIOPHANTINEoutperforms both DP++_ITER_RANGEand DECOMP. In addition, DIOPHANTINEspares the user of finding the optimum blow up factor and therefore does not need any pre-preprocessing either for optimizing the blow up factor or for the construction of a table unlike DP++_ITER_RANGEand DECOMP. We observe that performance of DP++_ITER_RANGEand DECOMPis indeed dependent on the choice of a blow up factor. It is difficult to propose the best value of a blow up factor either for DP++_ITER_RANGEor DECOMPsince, one would need to try all possible locally optimum values in a given range of B in order to determine the most favorable blow up factor.

From a theoretical standpoint, several outstanding questions remain open. The first one is related to the output sensitivity of DIOPHANTINE, which is observed in practice, but cannot be guaranteed in general. Getting finer insights on this phenomenon may allow developing more efficient enumeration schemes—by pruning sterile portions of the recursion tree. The second one is reminiscent from phase transitions, a well known phenomenon for selected hard optimization problems—such as 3-SAT for example. While the SD problem is about enumeration rather than optimization, linear redundancies within protein masses (one can trade a set of proteins against another set) have a direct impact on the rise of the number of solutions. A challenge is therefore to gain insights on the role of linear relationships and the total number of solutions. The third one is the asymptotic analysis of the number of solution of an interval stoichiometry determination

problem. While the asymptotics of the denumerant $D(M, 0)$ is known, working out the behavior or $D(M, \varepsilon)$ in conjunction with the distribution of solutions per node of the Hasse of the solution diagram sketch would provide valuable information on solutions as a function of the input masses and the noise level.

4.11 Supplemental: Implementation

4.11.1 Implementation sketch

Algorithms. Four algorithms were implemented in (generic) C++:

- The algorithm of [BL05b] to compute Frobenius numbers.
- The FPTAS of [GKM+11] to estimate the number of solutions of any knapsack problem.
- The algorithms `DIOPHANTINE` and `DP++` from section 4.3 to solve the interval SD problem.
- The algorithm of [BL05a] to solve SSP, called `DECOMP`. Recall that this algorithm has a pre-processing step consisting of computing the so-called *extended residue table* (ERT), followed by the backtracking step which decomposes a particular mass. To solve the interval SD problem, we compute the ERT once, and call the decomposition procedure for every mass in the interval.
- The rounding error algorithm of [BLLP06] to convert real number specification into an equivalent integer specification.
- The algorithm of [DLMB13] for determining the locally optimum blow up factor, b_{opt}.

The first three are made available within the program `addict`, which can be downloaded form `http://team.inria.fr/abs/addict/`.

To avoid overflows, unsigned integers coded on 64 rather than 32 bits were used. Such number types respectively accommodate integers up to $2^{64} - 1 \sim 0.18 \times 10^{20}$ and $2^{32} - 1 \sim 0.42 \times 10^{10}$, and instances with more than 10^{10} solutions are commonly faced—see Experiments. We note that on Linux operating systems, the type `unsigned long long` is always coded on 8 byte, be the system 32 or 64 bits, see e.g. `http://en.cppreference.com/w/cpp/language/types`.

For float type problems, the number type used to store values is double.

4.12 Supplemental: Material

4.12.1 Detailed Description of Biological Complexes

Yeast 19S Proteasome lid. Proteasomes are protein assemblies involved in elimination of damaged or misfolded proteins and the degradation of short-lived regulatory proteins. They are found in all eukaryotic cells and archaea, and also in selected bacteria. The most common form of the proteasome is 26S, named after its sedimentation coefficient—expressed in Svedberg.

The 26S Proteasome, consists of one core particle corresponding to the degradation chamber (the 20S) and of two regulatory caps filtering the entry of the proteins (the 19S). The 19S sub-complex itself subdivides into two other subcomplexes, the base that binds directly to the 20S core particle and a peripheral lid. The latter is composed of 9 different protein types with a single copy for each type. It is involved in recognition of the polyubiquitin chain of the substrate and followed by it's deubiquitination. This substrate is then unfolded and translocated to the core particle for further degradation [STA+06].

For yeast, the measured mass of the intact 19S Proteasome lid complex is 376,151 ± 369 Da. Summing the theoretical weights of the protein types amounts to 374,576 Da, i.e. the error in the measurement is 0.42%.

COP9 Signalosome. The COP9 Signalosome is a multifunctional complex primarily involved in ubiquitin mediated proteolysis linked to diverse cellular activities such as signal transduction, cell cycle progression and transcriptional regulation. It is composed of 8 protein types with a single copy for each type and shares remarkable homology with Yeast 19S proteasome lid complex.

An Electrospray MS experiment conducted on human COP9 signalosome reconstituted by coexpression in *E.coli* reveals the molecular weight for an intact complex to be 321,274 ± 35 Da. Summing the theoretical weights of the protein types amounts to 321,270 Da, i.e. the error in the measurement is 0.0012% [SMBE+09].

Eukaryotic Translation factor EIF3. Eukaryotic translation initiation factors (EIFs) are the proteins involved in assembling of elongation competent 80S ribosome to initiate the translation process. There are atleast nine EIFs involved in the initiation process. They carry out their function in two steps: formation of 48S complex with established codon-anticodon base pairing in the P-site of 40S ribosomal subunits, and the joining of 48S complex with 60S subunits [JHP10].

Among them, EIF3 binds to the 40S subunit of the ribosome to initiate protein synthesis followed by recruitment of messenger RNAs. It also promotes attachment of 43S complexes to mRNA. It has 13 different protein types with a single copy for each type. These 13 protein types have been unambiguously identified when MS/MS spectra, and were scanned against UniProt/Swiss-Prot and NCBInr database using the MASCOT search engine [DFZ+07]. The measured mass of the intact Yeast EIF3 complex is 797,999 ± 180 Da. Summing the theoretical weights of the protein types amounts to 793,558 Da, i.e. error w.r.t. to the theoretical weights is 0.56% [ZSF+08].

Yeast Exosome. The exosome is a 3'- 5' exonuclease complex involved in RNA processing and degradation. In eukaryotes it is present in the cytoplasm and nucleolus and therefore reacts with different substrates in respective compartments. It is composed of 10 different protein types each with unit stoichiometry [HDT+06]. The yeast exosome complex is known to contain domains homologous to ribonucleases e.g. RNase PH and RNase II and others, e.g. S1, KH, PINc and HRDC [ACL+02].

The measured mass of the intact Yeast exosome complex is 397,860 ± 99 Da. Summing the theoretical weights of the protein types amounts to 397,881 Da, i.e. error w.r.t. to the theoretical weights is 0.005%.

Rotary ATPases. Rotary ATPases are membrane associated molecular machines involved in energy conversion by coupling ATP hydrolysis (or synthesis) with proton (or Na^+) translocation across biological membranes. There are primarily two types of ATPases, F-type and V-type complexes each having two domains F_0, F_1 and V_0, V_1. The membrane embedded domains F_0/V_0 mediate proton (or Na^+) translocation and F_1/V_1 are involved in ATP production or consumption respectively.

An electrospray mass spectrum is recorded for rotary ATPases from *E. hirae* (EhATPase) and *E. thermus* (TtATPase). Each of these complexes have nine different protein types but, the membrane embedded rotor for EhATPase is larger because each K type contains four transmembrane helices as compared to two transmembrane helices in corresponding L type in TtATPase.

MS experiment data was retrieved from [MR12]. EhATPase was undergone controlled disassembly resulting in formation of sub-complexes in gas phase and solution phase using collision induced dissociation (CID) and partial denaturation by manipulating the ionic strength, respectively. We choose four sub-complexes – 2,3,4 and 5 formed in the solution phase, and assumed that each subcomplex is composed of all the 9 types of proteins as does the intact complex. Measured molecular weights of subcomplexes and percentage error in measurement are shown in table 4.10.

Complex	Stoichiometries	Measured Mass (Da)	$\sum s_i w_i^t$	%error in measurement
EhATPase-sub-2	$A_3B_3CDE_2F_2G$	500,178 ± 294	499,131	0.21
EhATPase-sub-3	$A_3B_3DE_2F_2G$	461,674 ± 324	460,968	0.15
EhATPase-sub-4	A_3B_3DEFG	424,441 ± 148	423,813	0.15
EhATPase-sub-5	A_3B_3DG	387,356 ± 230	386,658	0.18
TtATPase	$A_3B_3CDE_2G_2FIL_{12}$	659,202 ± 131	657,979	0.19

Table 4.10: Measured molecular weights (in Da) and percentage error in measurement w.r.t. weighted sum of the theoretical masses of individual protein types complying with their known stoichiometries. Refer to fig. 6 of [MR12] and Table S1A and S1B of supplementary information of [ZMB+11]

Yeast Nuclear Pore Complex (NPC). The NPC is a protein assembly anchored in the nuclear envelope, regulating the nucleo-cytoplasmic transport. It is composed of $\sim$ 30 distinct proteins types each present in

multiple copies, and is the largest protein assembly known to date in the eukaryotic cell [WR10, DH08]. It has eight-fold radial symmetry and consists of eight spokes, each with a cytoplasmic and a nuclear side.

While we are not aware of mass spectrometry experiments on the whole NPC, a number of sub-systems have been studied in detail. One of them is an heptameric sub-complex known as the Y-complex [FMPS$^+$12, DDC12], which participates to the formation of the scaffold of the NPC, as one finds one copy of the Y-complex per half-spoke.

In the sequel, we simulated mass spectrometry data for two complexes. The first one is a complete spoke, which contains a total of 57 proteins instances of 30 different types. The second one is the Y-ring complex, containing 8 copies of the Y-complex—that is 56 proteins of 7 types. For each complex, noise levels of 0%, 0.1% and 1% were applied to the exact mass of the complex computed from the masses of the individual proteins.

4.13 Supplemental: Algorithm `DIOPHANTINE`

In computer science, an algorithm is termed *output sensitive* when its time complexity is a function of the size of the output. Since the number of solutions of a SD problem might be large, we precisely investigate the output sensitivity of `DIOPHANTINE`.

4.13.1 Output Sensitive Behavior for Integer Type Problems

Rationale. In the exact case, `DIOPHANTINE` can only be output sensitive for a target mass beyond the Frobenius number, since any call with a non representable mass necessarily results in useless operations. The running times of `DIOPHANTINE` being comparable to those of the output sensitive algorithm `DP++`, we analysis the output sensitive behavior of `DIOPHANTINE`.

Results: running time versus number of non-negative solutions. We first computed the Pearson correlation coefficient between $t^{\text{Tot}}_{\texttt{DIO}}$ and the number of solutions (Fig. SI-4.9). With values equal to 0.97, 0.89 and 0.97 for the three datasets at 0.1% noise level, and to 0.97, 0.89 and 0.97 at 1% noise level, this correlation is excellent.

Results: running time versus the recursion tree size. The correlation between $t^{\text{Tot}}_{\texttt{DIO}}$ and the recursion tree size is even better (Fig. SI-4.10), with coefficients of 0.99, 0.99 and 0.99 at 0.1% noise level, and of 0.99, 0.95 and 0.98 at 1% noise level, respectively for the three datasets.

We further plotted the proportion of fertile edges as a function of the target mass (Fig. SI-4.11). Note that this proportion is 1 when the condition of Eq. (4.25) holds, which is the case for 0/410 and 338/410 instances at 0.1% and 1% respectively. As expected, this proportion increases with the error level: at 1%, all the instances which do not meet the condition of Eq. (4.25) have a ratio above 0.6, while no instance has such a high ratio at 0.1%.

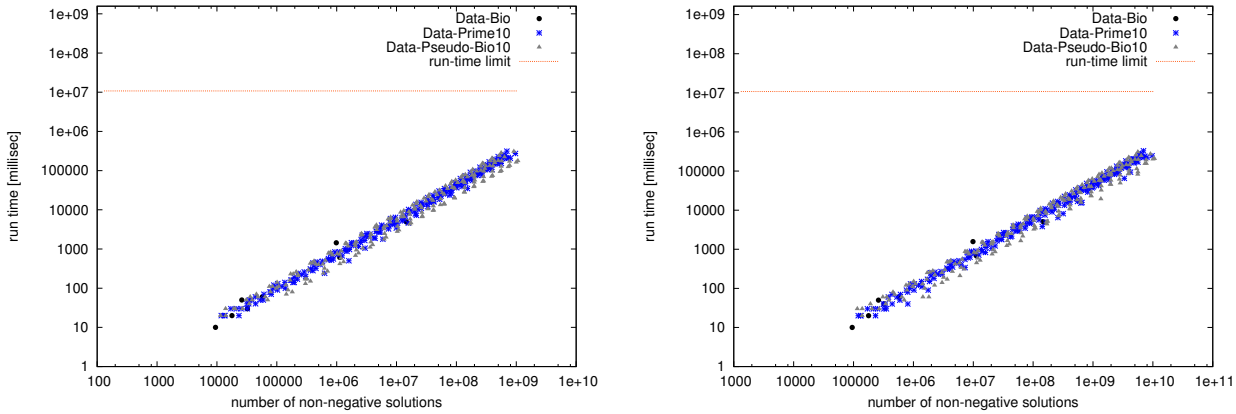


Figure 4.9: `DIOPHANTINE`: **running time $t^{\mathbf{Tot}}_{\mathtt{DIO}}$ as a function of the number of solutions.** (Left) Noise level of 0.1% (Right) Noise level of 1%. *Note that Y-axes are drawn with a logarithmic scale.*

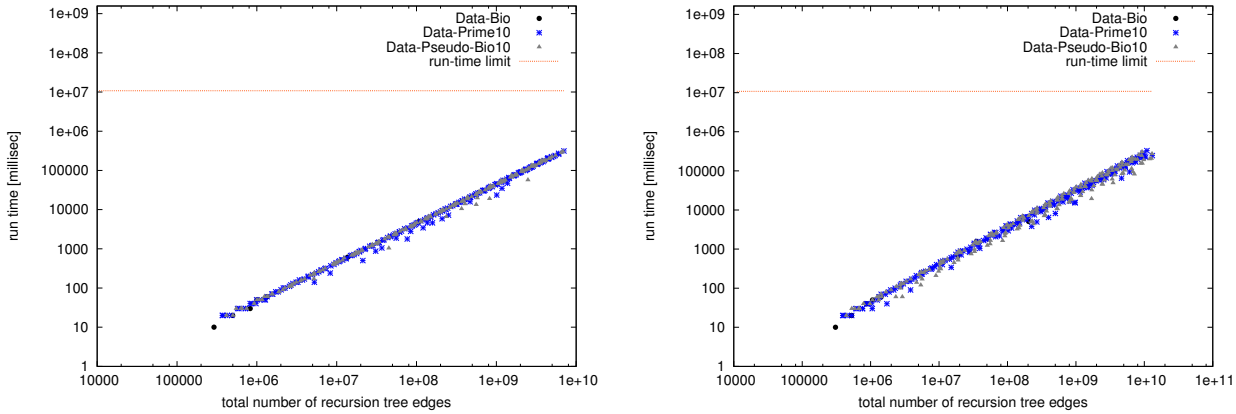


Figure 4.10: `DIOPHANTINE`: **running time $t^{\mathbf{Tot}}_{\mathtt{DIO}}$ as a function of the recursion tree size.** (Left) Noise level of 0.1% (Right) Noise level of 1%. *Note that Y-axes are drawn with a logarithmic scale.*

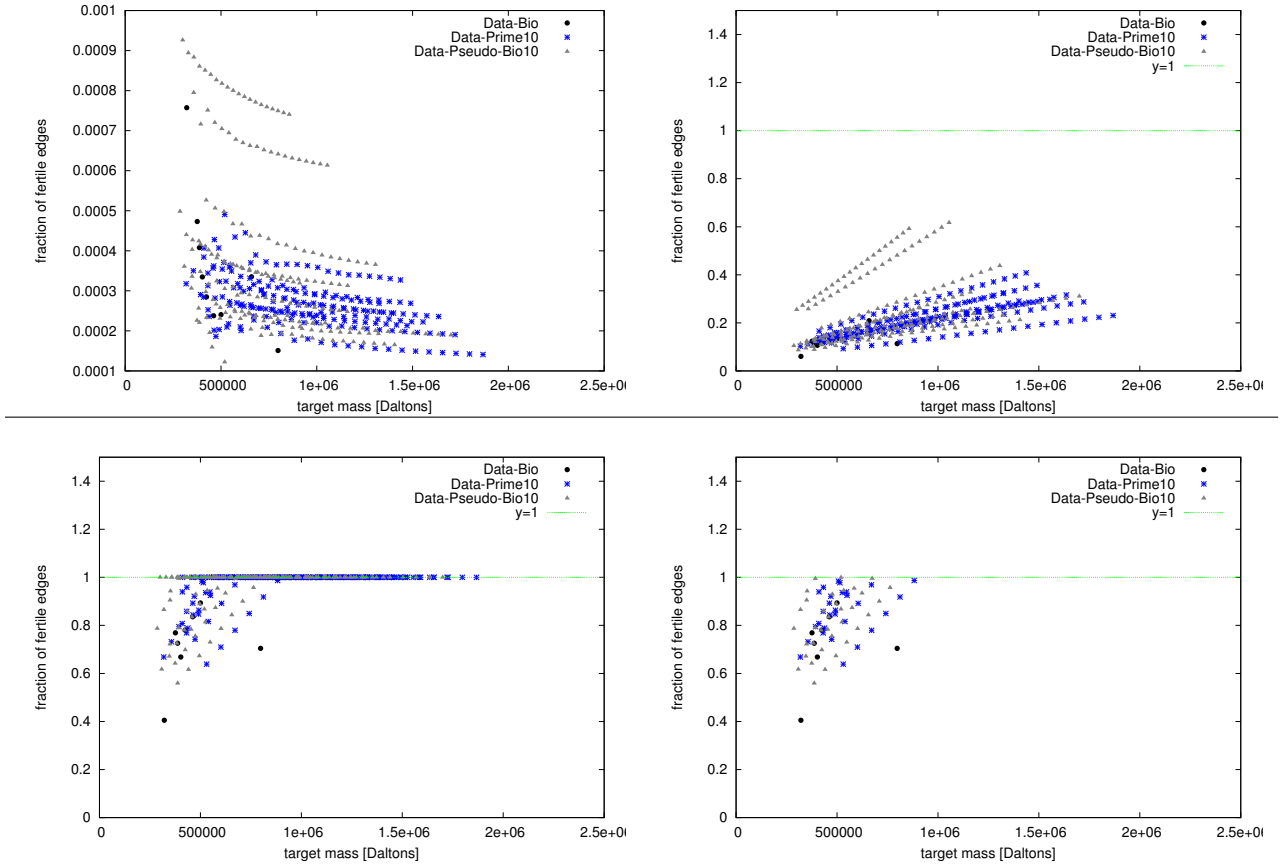


Figure 4.11: DIOPHANTINE: **fraction of fertile edges at three noise levels, 0% (Top-Left), 0.1% (Top-Right), 1%(Bottom-Left).** The last figure pertains to those instances with 1% noise, which do not meet with the output sensitivity criterion stated in Eq. (4.25).

4.14 Supplemental: Algorithm DP++

4.14.1 Float Type Problems

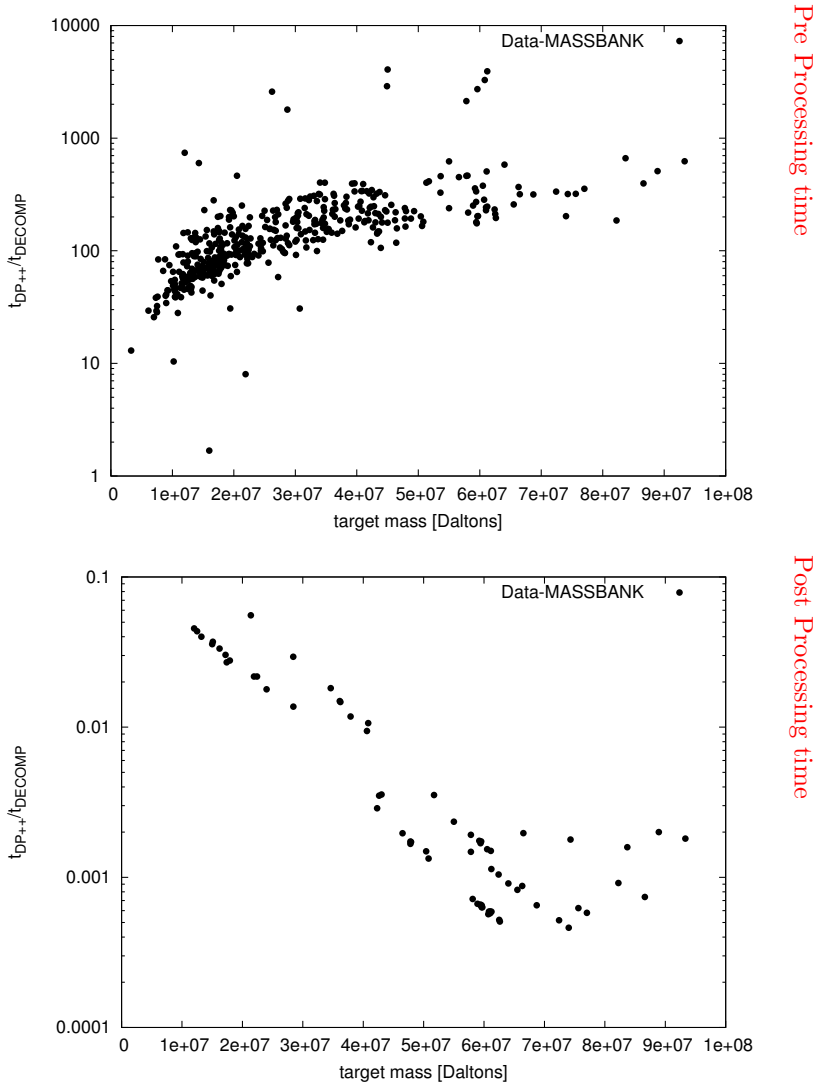


Figure 4.12: **On the MassBank dataset DP++is outperformed by DECOMPdue to slower pre-processing step (Top), although the post-processing step is substantially fast (Bottom).**

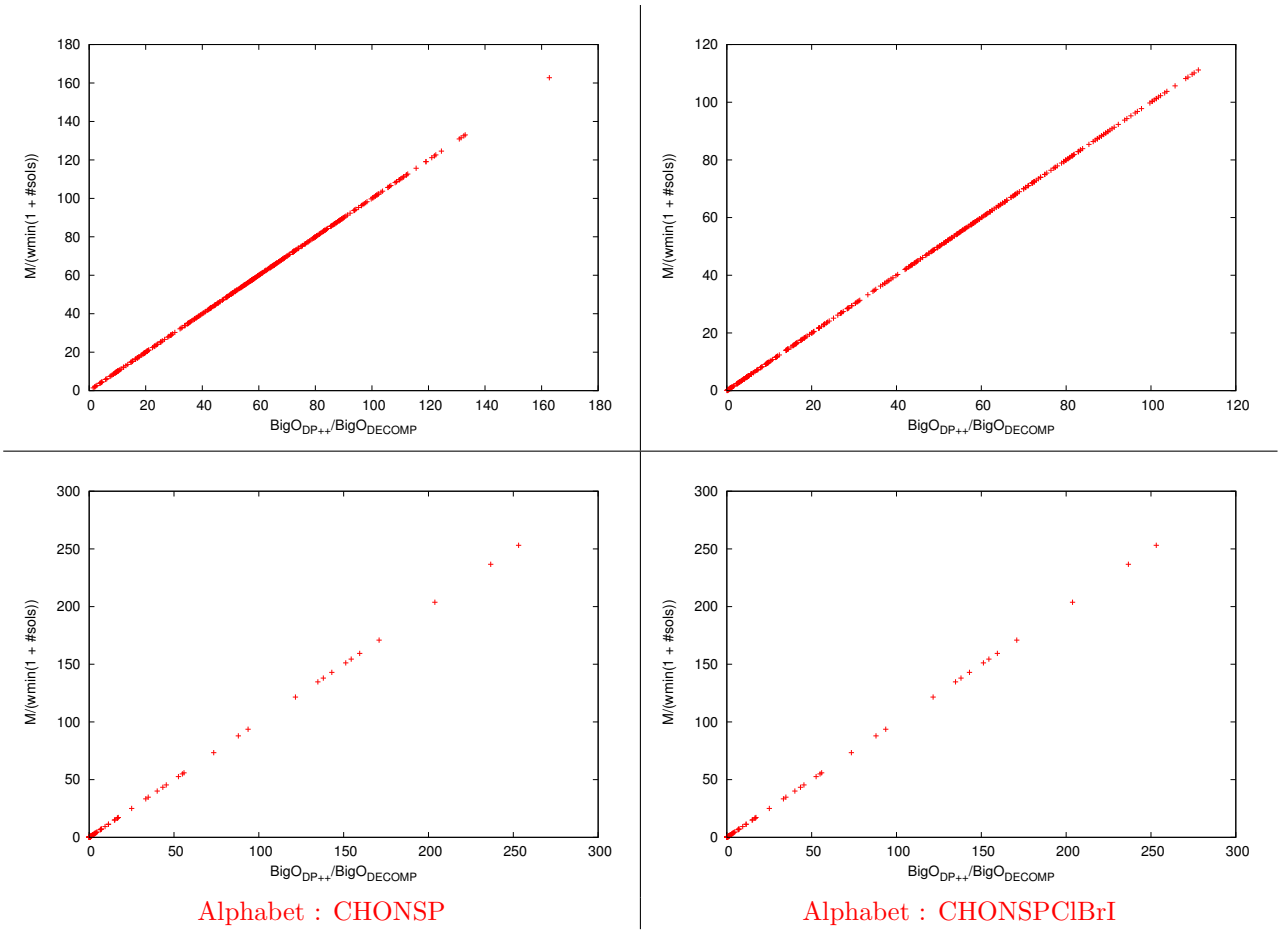


Figure 4.13: **The excellent correlation suggests that the ratio not only depends upon the target mass and min weight but also on the number of solutions.** Plots corresponds to the MASSBANK and Synthetic Metaolites decomposed over two alphabet sizes. Noise level is 20ppm.

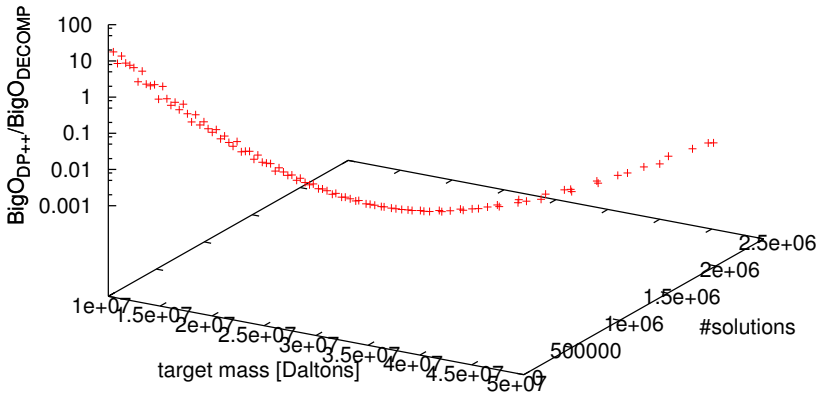


Figure 4.14: `Data-SynMetab9`; B=30000

4.14.2 Integer Type Problems

Decreasing performance of `DECOMP`

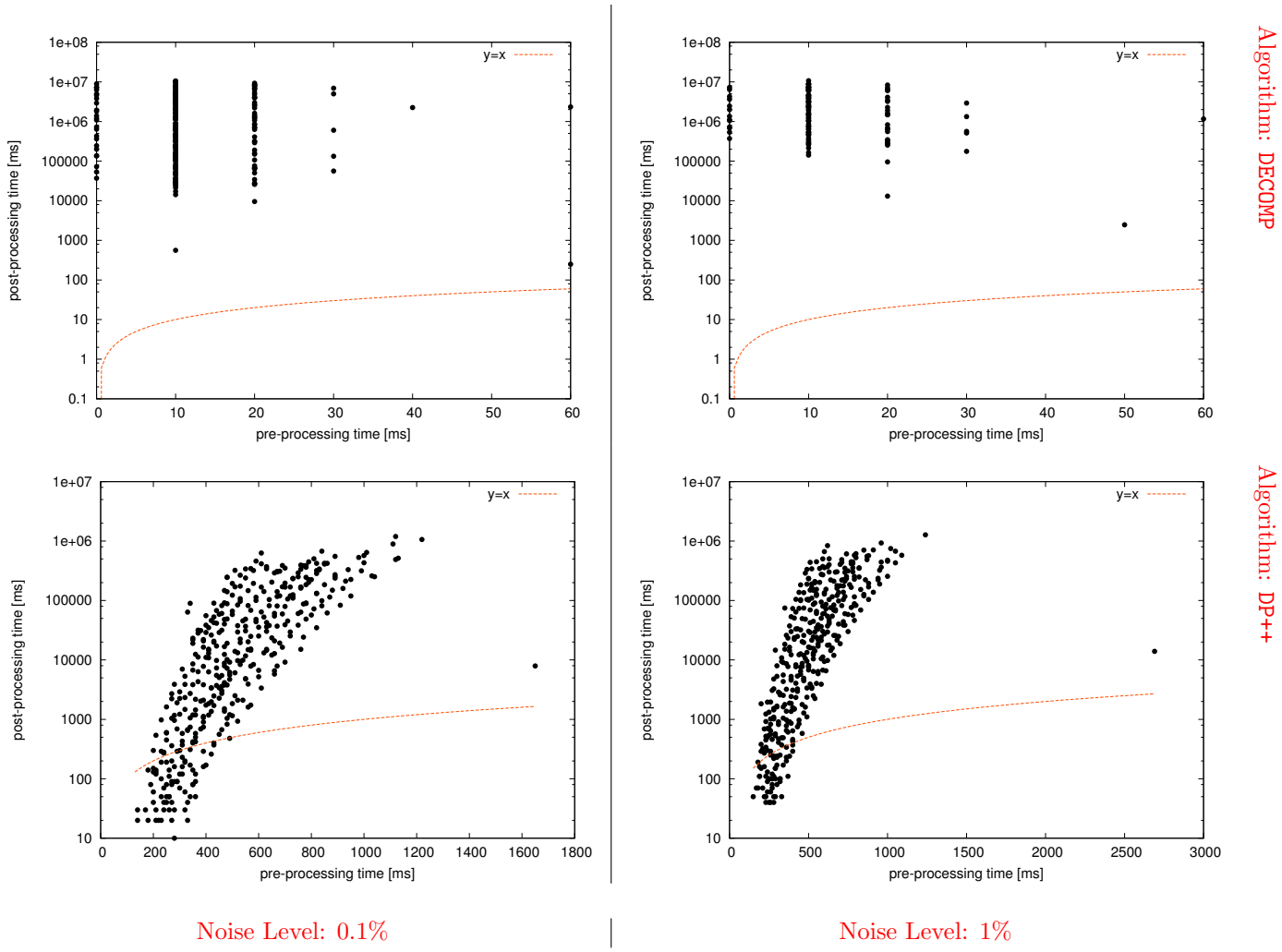
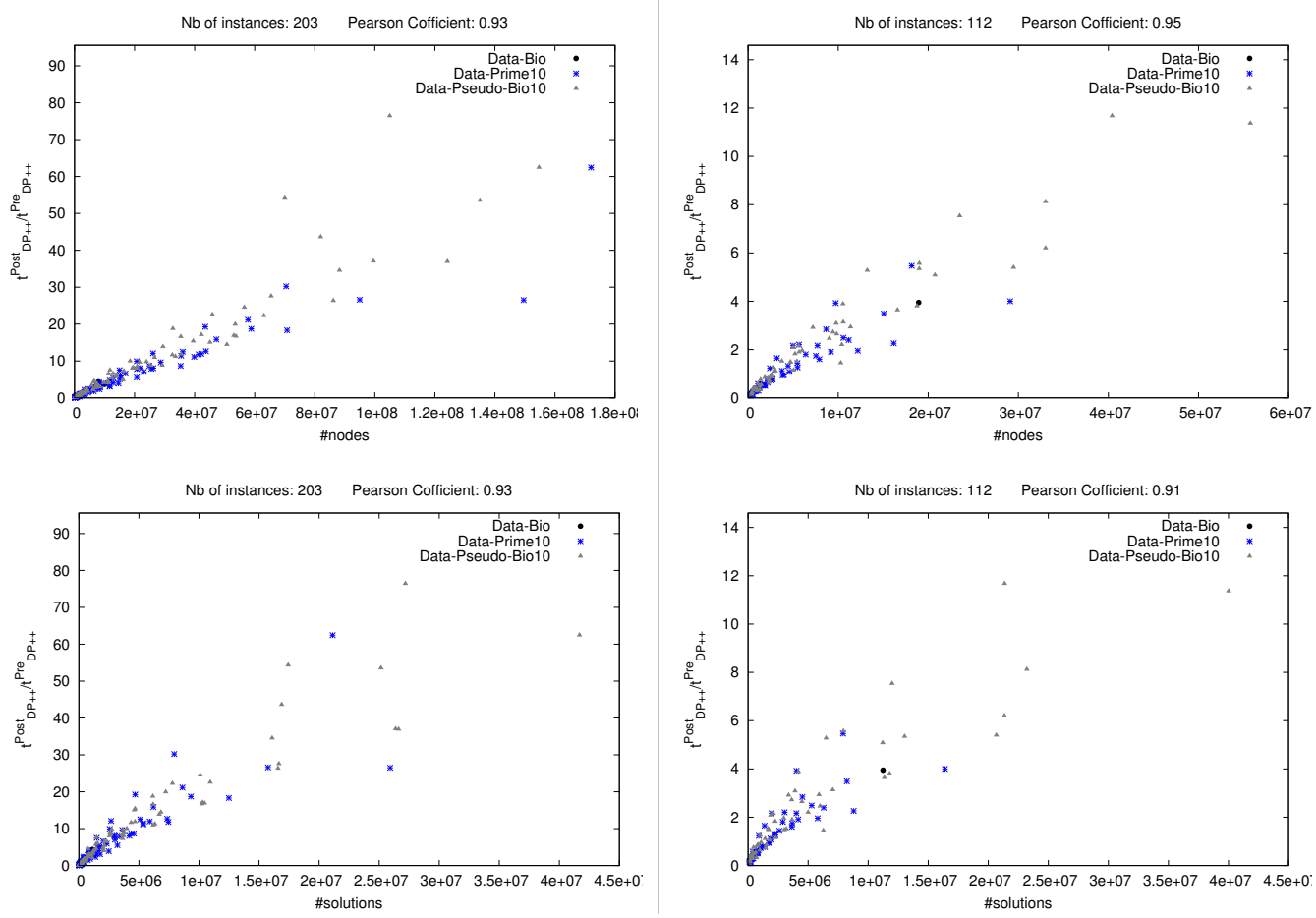


Figure 4.15: DECOMP **[Top] and** DP++**[Bottom]: post-processing time (i.e.** $t^{\mathbf{Post}}$**) as a function of the pre-processing time (i.e.** $t^{\mathbf{Pre}}$**).** *Note that Y-axes are drawn with a logarithmic scale.*

Table 4.11: DP++**: post-processing time (i.e.** $t^{\mathbf{Post}}_{\mathtt{DP++}}$**) dominates the pre-processing time (i.e.** $t^{\mathbf{Pre}}$**) for instances with large #Nodes and large #Solutions.** Numbers provided are Pearson correlation coefficient at 0.1% and 1% noise level, respectively.

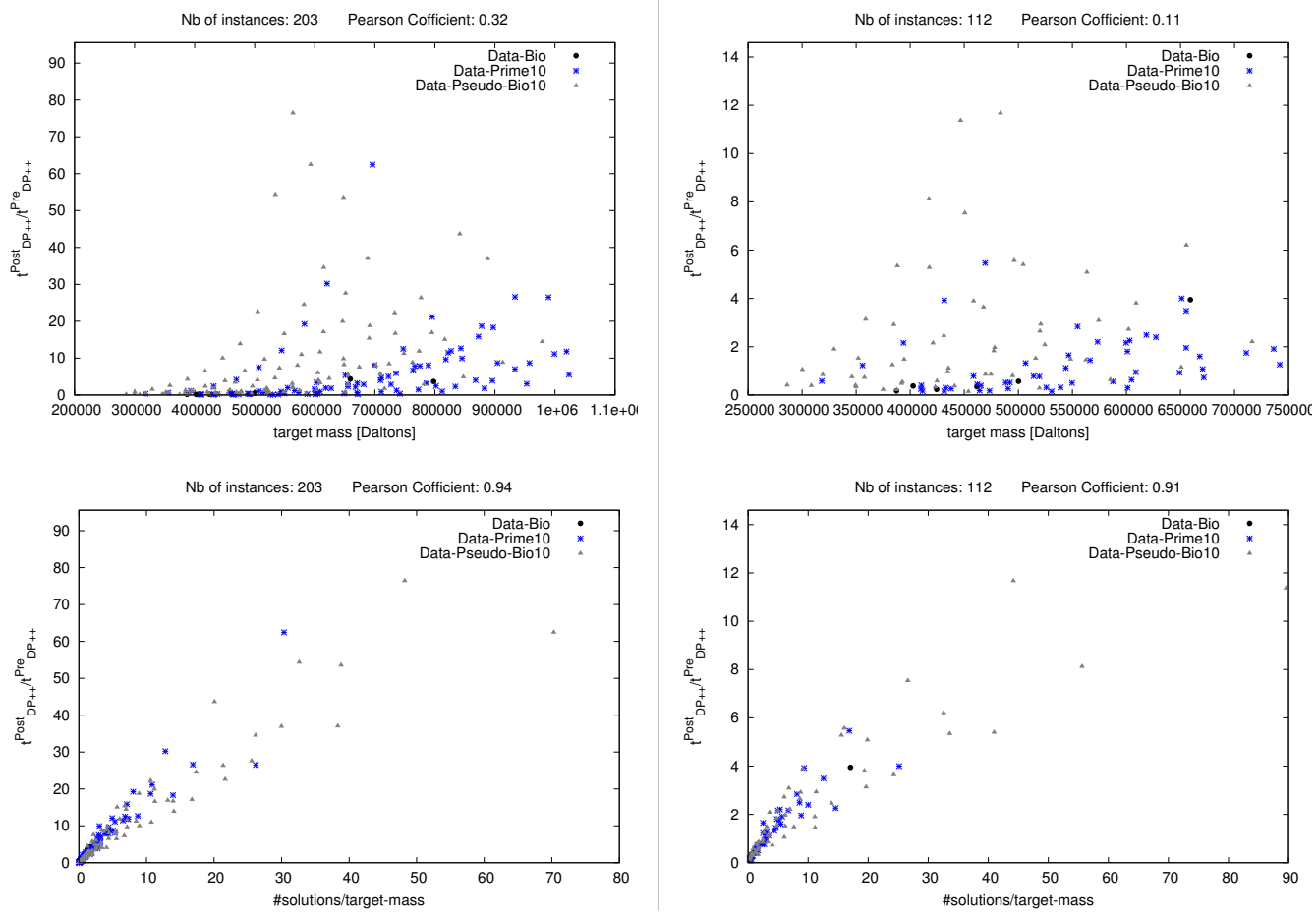
	#Nodes		*#Solutions*		*Target Mass*		$\frac{\#Solutions}{Target\ mass}$	
Noise level	0.1%	1%	0.1%	1%	0.1%	1%	0.1%	1%
$\frac{t^{\mathrm{Post}}_{\mathtt{DP++}}}{t^{\mathrm{Pre}}_{\mathtt{DP++}}}$	0.93	0.95	0.93	0.91	0.32	0.11	0.94	0.91

4.14.3 Plots Corresponding to the Table 4.11



Noise Level: 0.1% | Noise Level: 1%

Figure 4.16: DP++: **The Dominance of $t^{\mathbf{Post}}_{\mathtt{DP++}}$ over $t^{\mathbf{Pre}}_{\mathtt{DP++}}$ increases for instances having large tree size (#nodes) and large #solutions.** **[Top]** $t^{\mathrm{Post}}_{\mathtt{DP++}}/t^{\mathrm{Pre}}_{\mathtt{DP++}}$ as a function of number of nodes and **[Bottom]** Number of solutions, at 0.1% and 1% noise level for three datasets.



Noise Level: 0.1% | Noise Level: 1%

Figure 4.17: DP++: **Increase in target mass does not contribute in dominance of $t^{\mathbf{Post}}_{\mathtt{DP++}}$ over $t^{\mathbf{Pre}}_{\mathtt{DP++}}$.** **[Top]** $t^{\mathrm{Post}}_{\mathtt{DP++}}/t^{\mathrm{Pre}}_{\mathtt{DP++}}$ as a function of Target mass in Daltons and **[Bottom]** ratio of Number of solutions over Target mass, at 0.1% and 1% noise level for three datasets.

4.15 Supplemental: Algorithms DP++vs. DECOMP

4.15.1 Integer Type Problems

Table 4.12: **DP++ and DECOMP: increase in $t^{\textbf{Post}}_{\texttt{DECOMP}}$ over $t^{\textbf{Post}}_{\texttt{DP++}}$ is due to relative size of trees (#nodes) for both the algorithms.** Pairwise Pearson correlation coefficients to compare DECOMP and DP++ at two noise levels. (See also figs. SI-4.19 – SI-4.22).

Noise Level – 0.1%				
	$\frac{\#nodes_{\text{DECOMP}}}{\#nodes_{\text{DP++}}}$	$\frac{t^{\text{Tot}}_{\texttt{DECOMP}}}{t^{\text{Tot}}_{\texttt{DP++}}}$	$\frac{t^{\text{Pre}}_{\texttt{DECOMP}}}{t^{\text{Pre}}_{\texttt{DP++}}}$	$\frac{t^{\text{Post}}_{\texttt{DECOMP}}}{t^{\text{Post}}_{\texttt{DP++}}}$
error width	0.37	0.74	-0.21	0.16
$\frac{\#nodes_{\text{DECOMP}}}{\#nodes_{\text{DP++}}}$		0.48	0.46	0.75
$\frac{t^{\text{Tot}}_{\texttt{DECOMP}}}{t^{\text{Tot}}_{\texttt{DP++}}}$			-0.01	0.43
$\frac{t^{\text{Pre}}_{\texttt{DECOMP}}}{t^{\text{Pre}}_{\texttt{DP++}}}$				0.49

Noise Level – 1%				
	$\frac{\#nodes_{\text{DECOMP}}}{\#nodes_{\text{DP++}}}$	$\frac{t^{\text{Tot}}_{\texttt{DECOMP}}}{t^{\text{Tot}}_{\texttt{DP++}}}$	$\frac{t^{\text{Pre}}_{\texttt{DECOMP}}}{t^{\text{Pre}}_{\texttt{DP++}}}$	$\frac{t^{\text{Post}}_{\texttt{DECOMP}}}{t^{\text{Post}}_{\texttt{DP++}}}$
error width	0.51	0.66	-0.14	0.58
$\frac{\#nodes_{\text{DECOMP}}}{\#nodes_{\text{DP++}}}$		0.35	0.29	0.79
$\frac{t^{\text{Tot}}_{\texttt{DECOMP}}}{t^{\text{Tot}}_{\texttt{DP++}}}$			-0.13	0.73
$\frac{t^{\text{Pre}}_{\texttt{DECOMP}}}{t^{\text{Pre}}_{\texttt{DP++}}}$				0.12

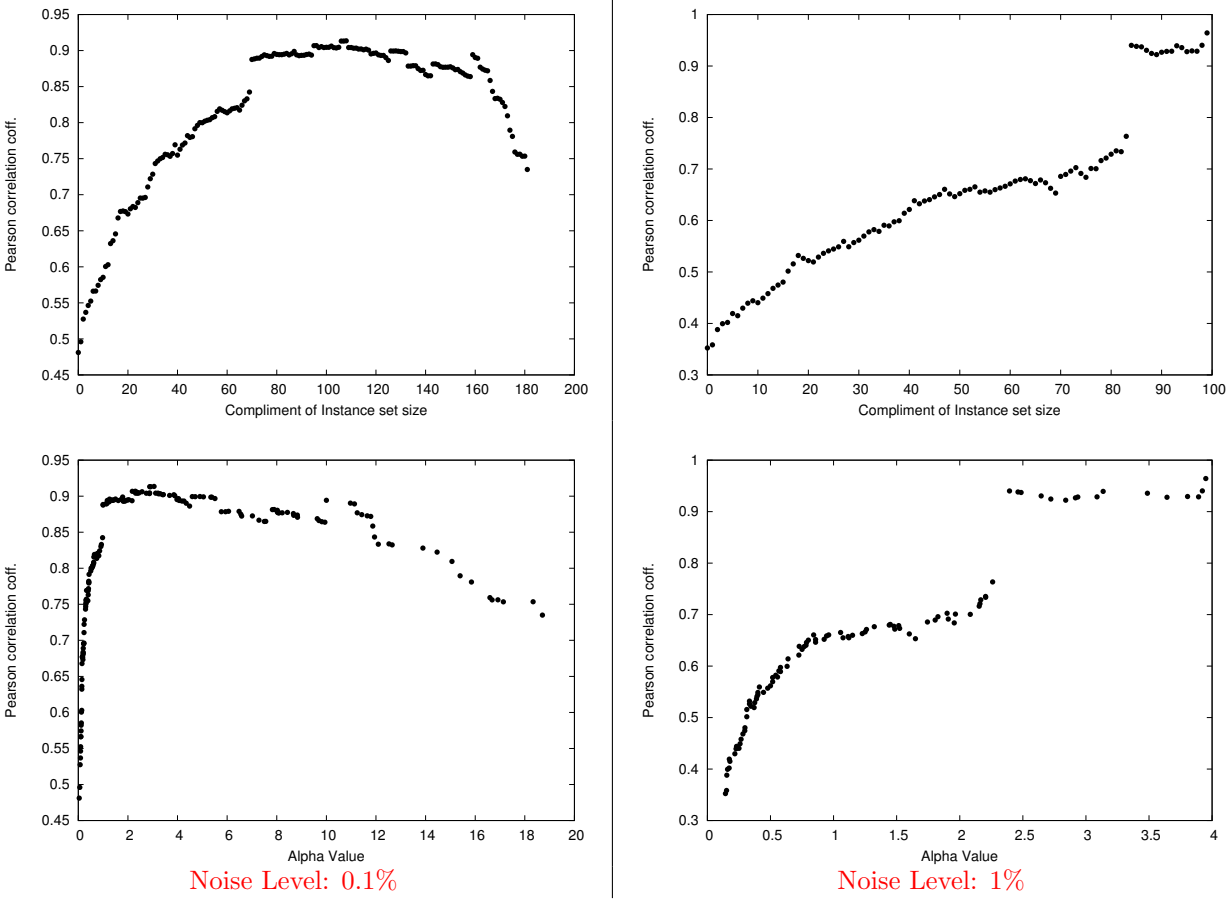


Figure 4.18: **The parameterized correlation between the ratios of total running times $t^{\mathbf{Tot}}_{\texttt{DECOMP}}/t^{\mathbf{Tot}}_{\texttt{DP++}}$ and $\#nodes_{\mathbf{DECOMP}}/\#nodes_{\mathbf{DP++}}$ increases upon stepwise removal of instances. (Top)** At an x-coordinate $= k$, the Pearson correlation value corresponds to the $N - k$ instances of the set $S_{>i}$, see Eq. (4.37). **(Bottom)** The x-axis features the sorted α_i values.

4.15.2 Plots Corresponding to the Table 4.12

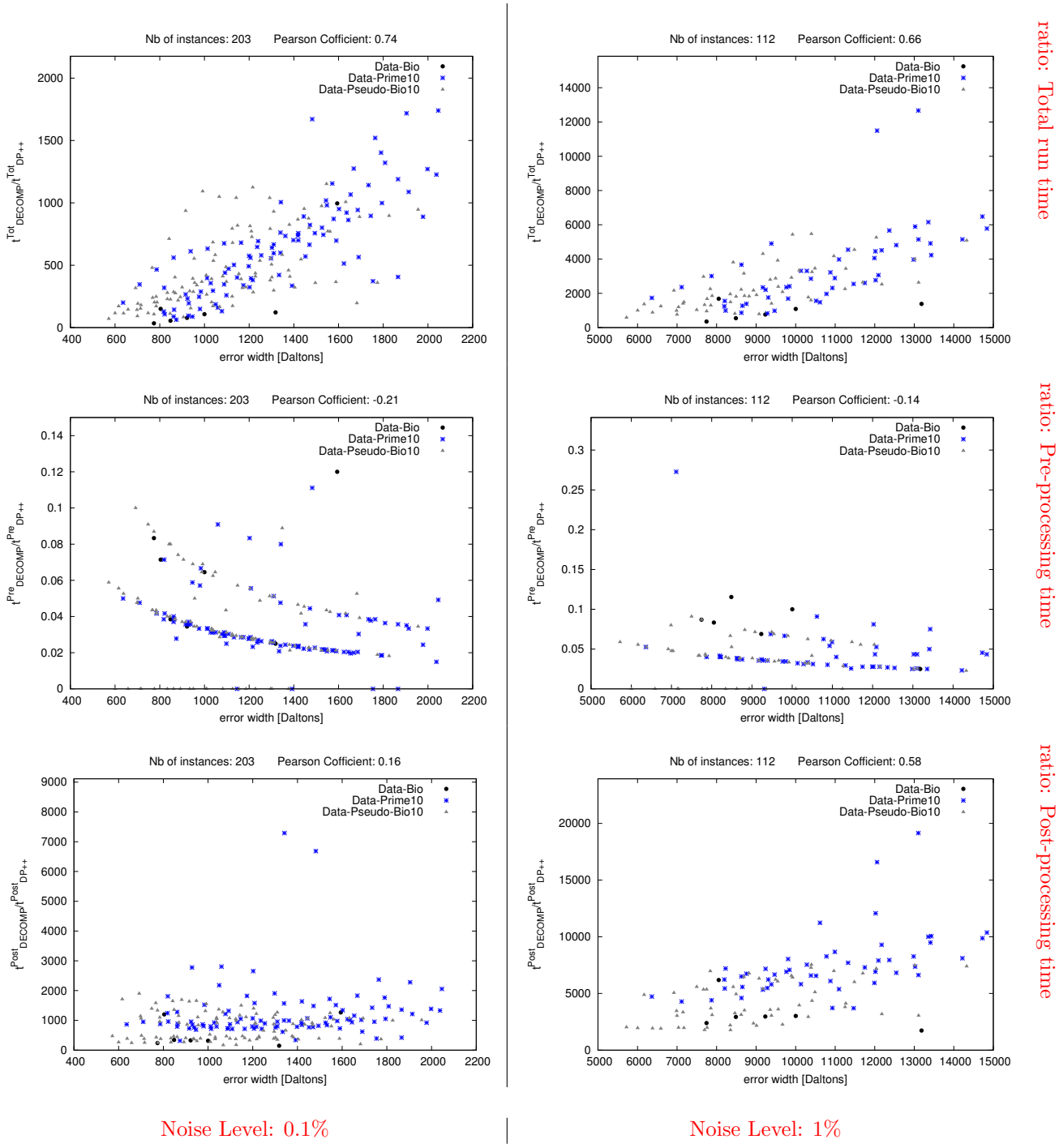


Figure 4.19: **Pairwise plots between error width and ratio of run times for DECOMP over DP++. (Columns)** The left and right columns respectively correspond to noise levels 0.1% and 1%. **(Top row)** Error width vs. Ratio of total run time for DECOMP over DP++ **(Middle row)** Error width vs. Ratio of post-processing time (backtracking) for DECOMP over DP++ **(Bottom row)** Error width vs. Ratio of pre-processing time for DECOMP over DP++

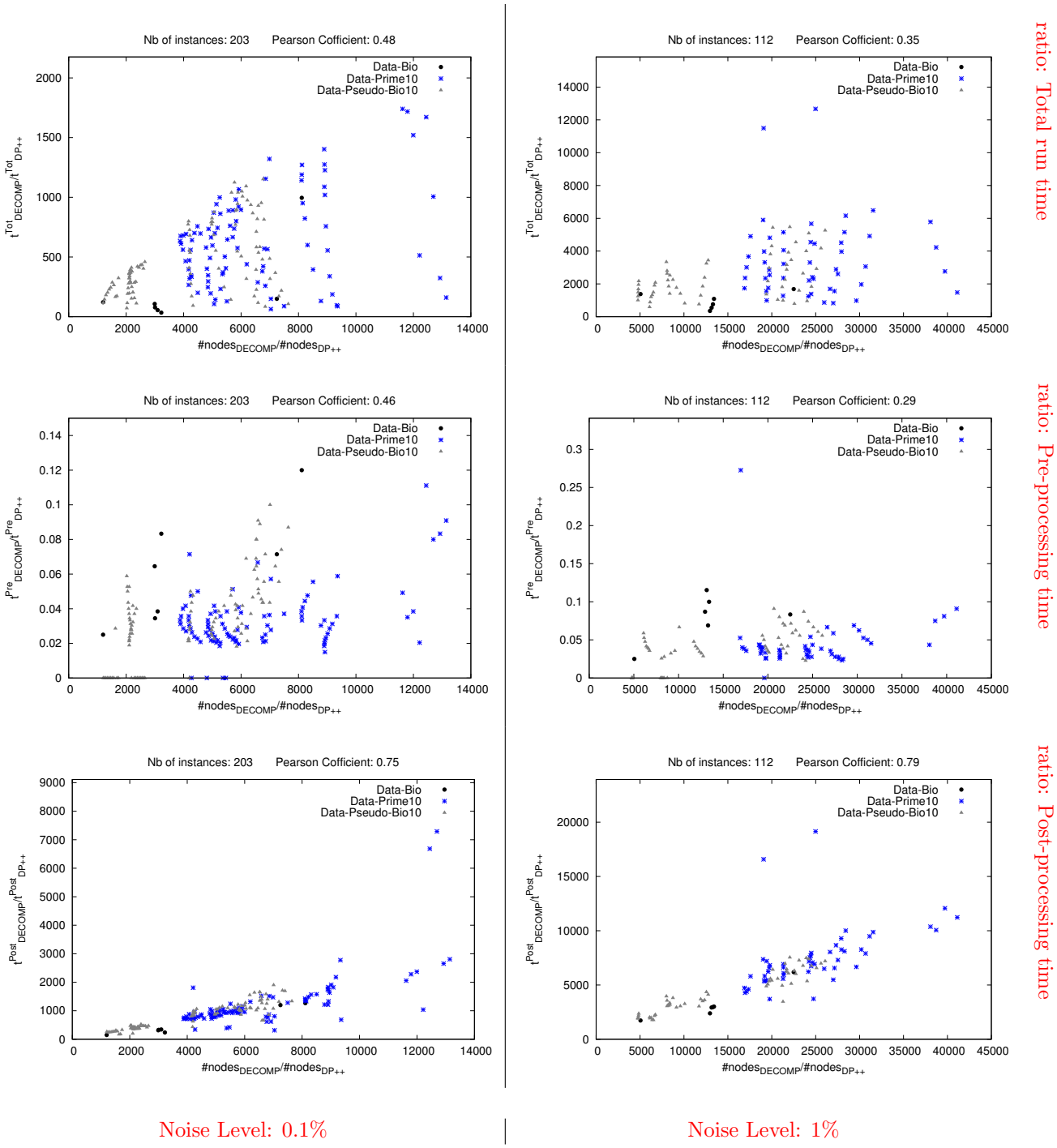
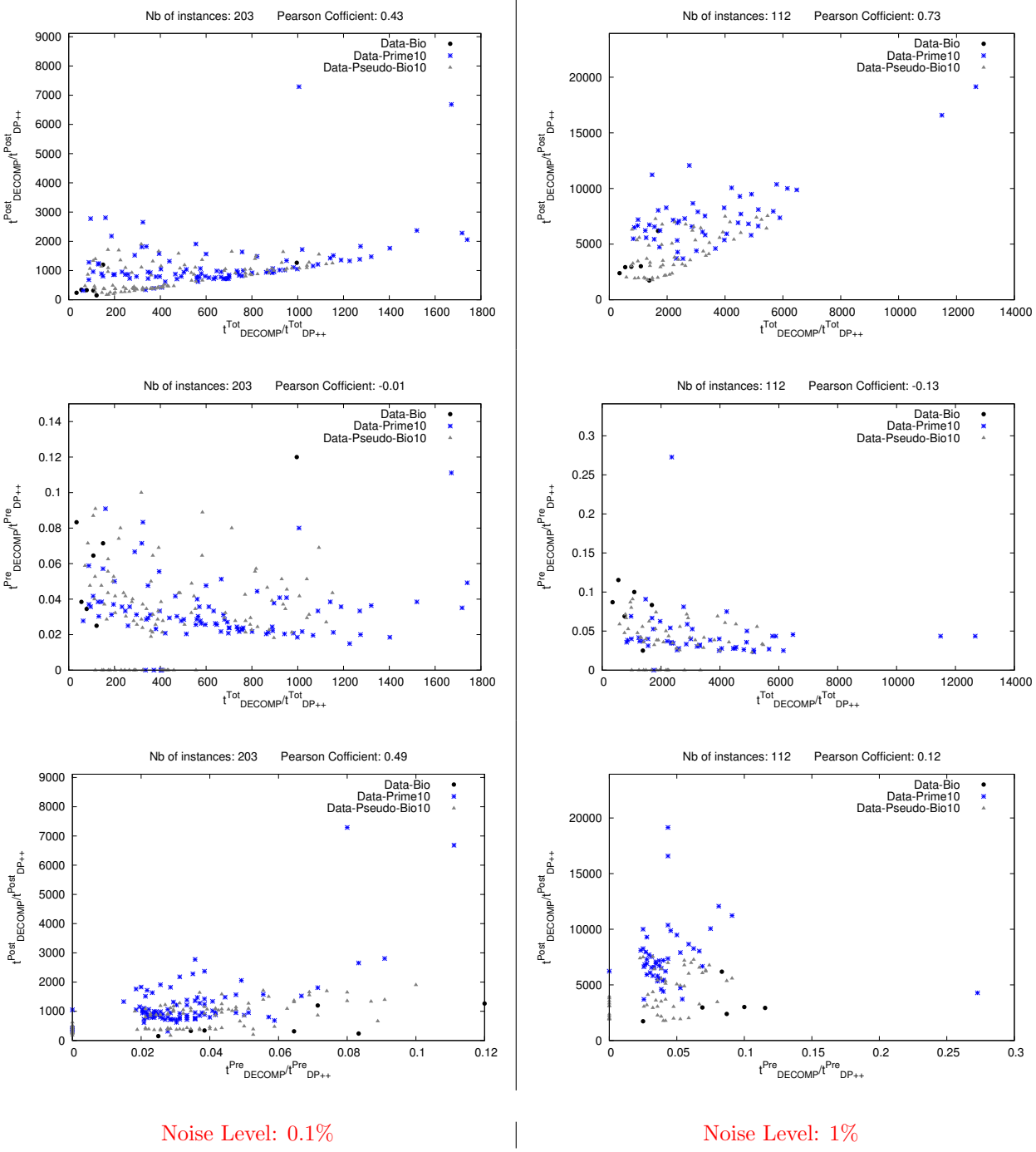


Figure 4.20: **Pairwise plots among ratio of runtimes and ratio of number of nodes for** `DECOMP` **over** `DP++` **(Columns)** The left and right columns respectively correspond to noise levels 0.1% and 1%. **(Top row)** Ratio of number of nodes explored vs. Ratio of total run time for `DECOMP` over `DP++` **(Middle row)** Ratio of number of nodes explored vs. Ratio of post-processing time (backtracking) for `DECOMP` over `DP++` **(Bottom row)** Ratio of number of nodes explored vs. Ratio of pre-processing time for `DECOMP` over `DP++`



Noise Level: 0.1% | Noise Level: 1%

Figure 4.21: **Pairwise plots among ratio of total run times, post-processing time and pre-processing time for DECOMP over DP++. (Columns)** The left and right columns respectively correspond to noise levels 0.1% and 1%. **(Top row)** Ratio of total run time vs. Ratio of post-processing time (backtracking) DECOMP over DP++ **(Middle row)** Ratio of total run time vs. Ratio of pre-processing time for DECOMP over DP++ **(Bottom row)** Ratio of post-processing time (backtracking) vs. Ratio of post-processing time for DECOMP over DP++

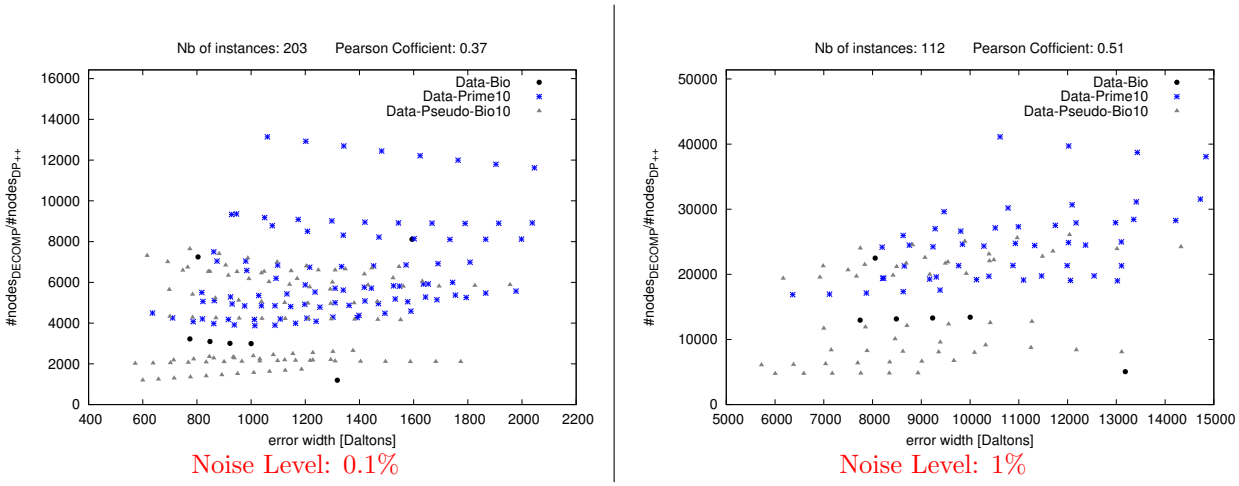


Figure 4.22: Error width vs. Ratio of number of nodes explored for `DECOMP` over `DP++`.

4.15.3 Studying the hierarchical tree for Biological complexes

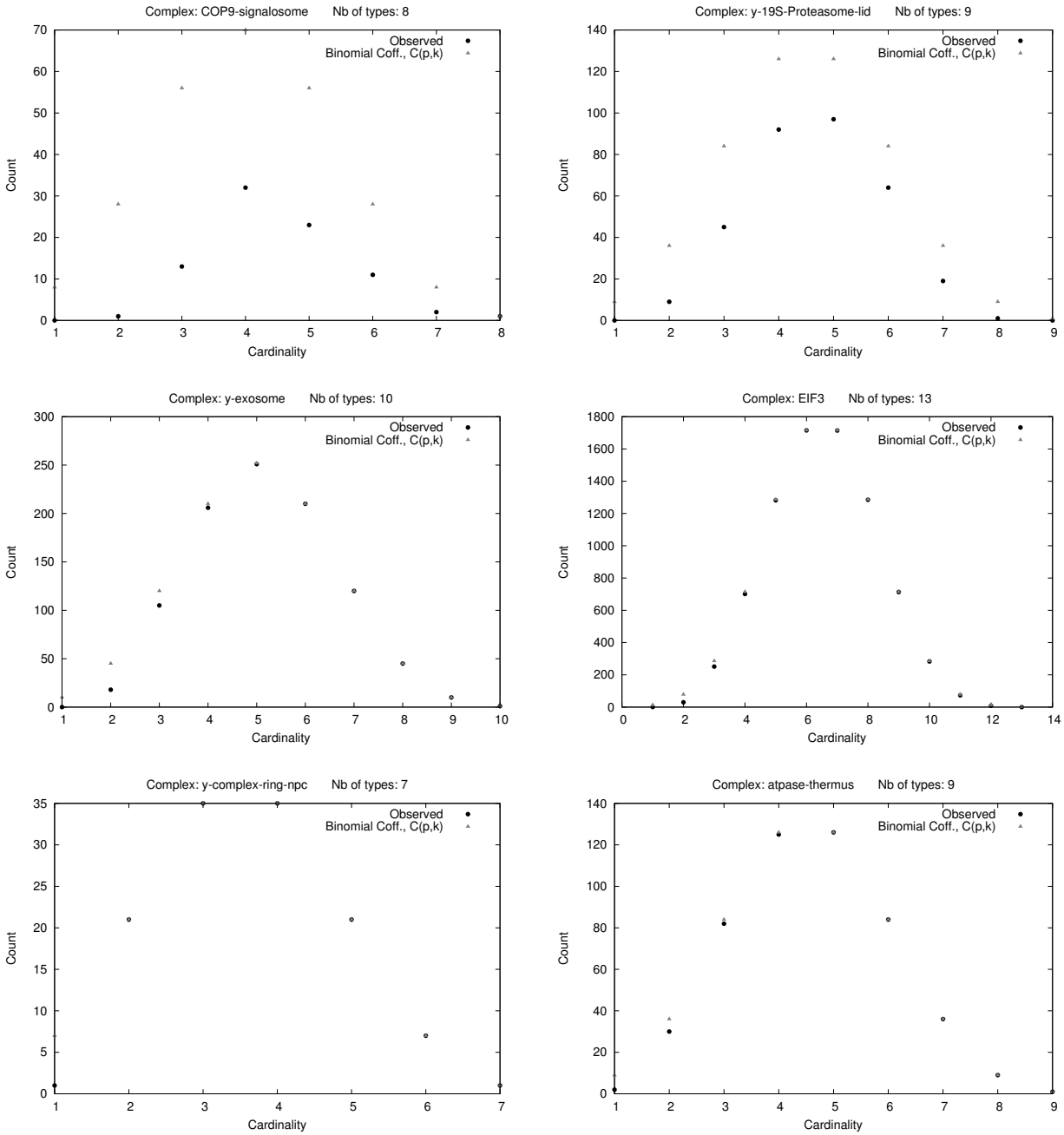


Figure 4.23: **Number of tuples of different cardinality used in the Biological complexes are close to the respective Binomial coefficients even at small noise level**. Cardinality or size of the tuple corresponds to the non-negative solutions of 0.1%.

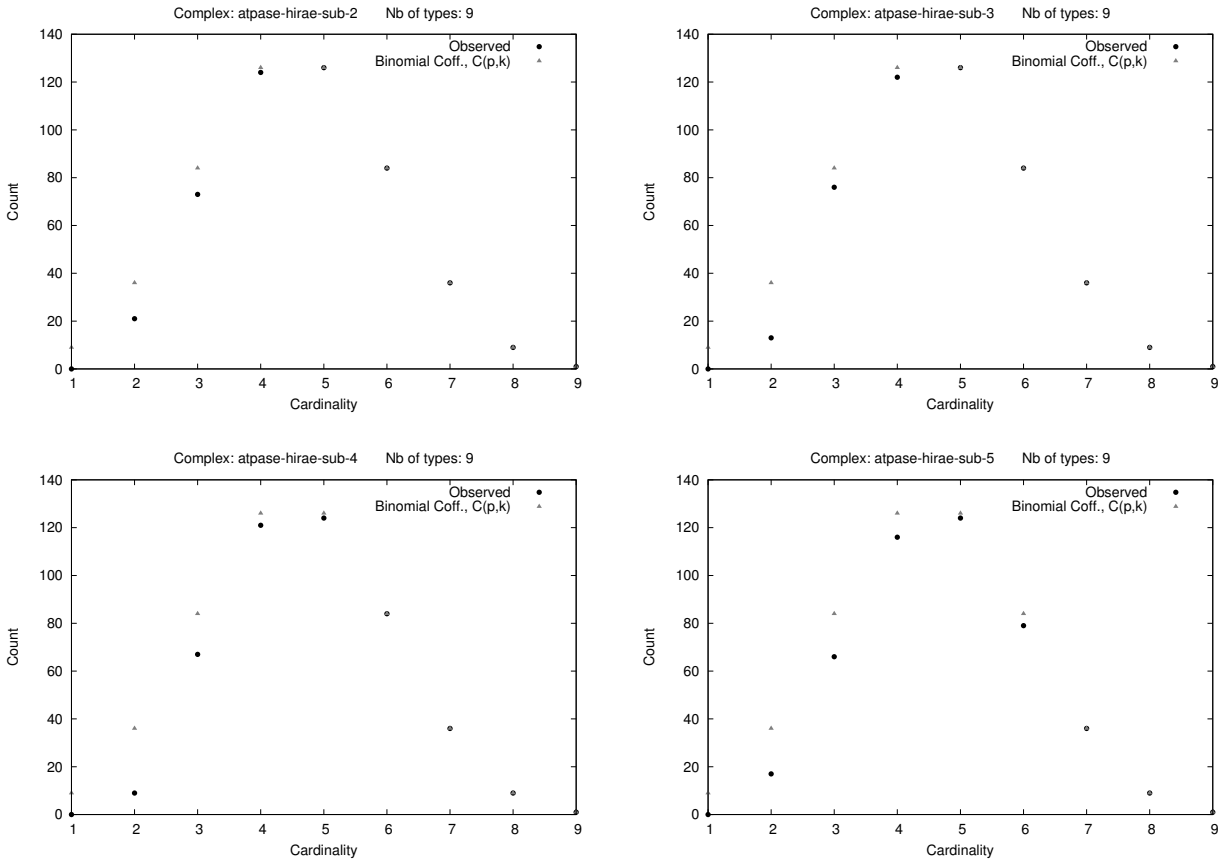


Figure 4.24: **Number of tuples of different sizes used in the Biological complexes are close to the respective Binomial coefficients even at small noise level**. Cardinality or size of the tuple corresponds to the non-negative solutions at noise level of 0.1%.

4.15.4 Scatter plots to compare DP++ and DECOMP

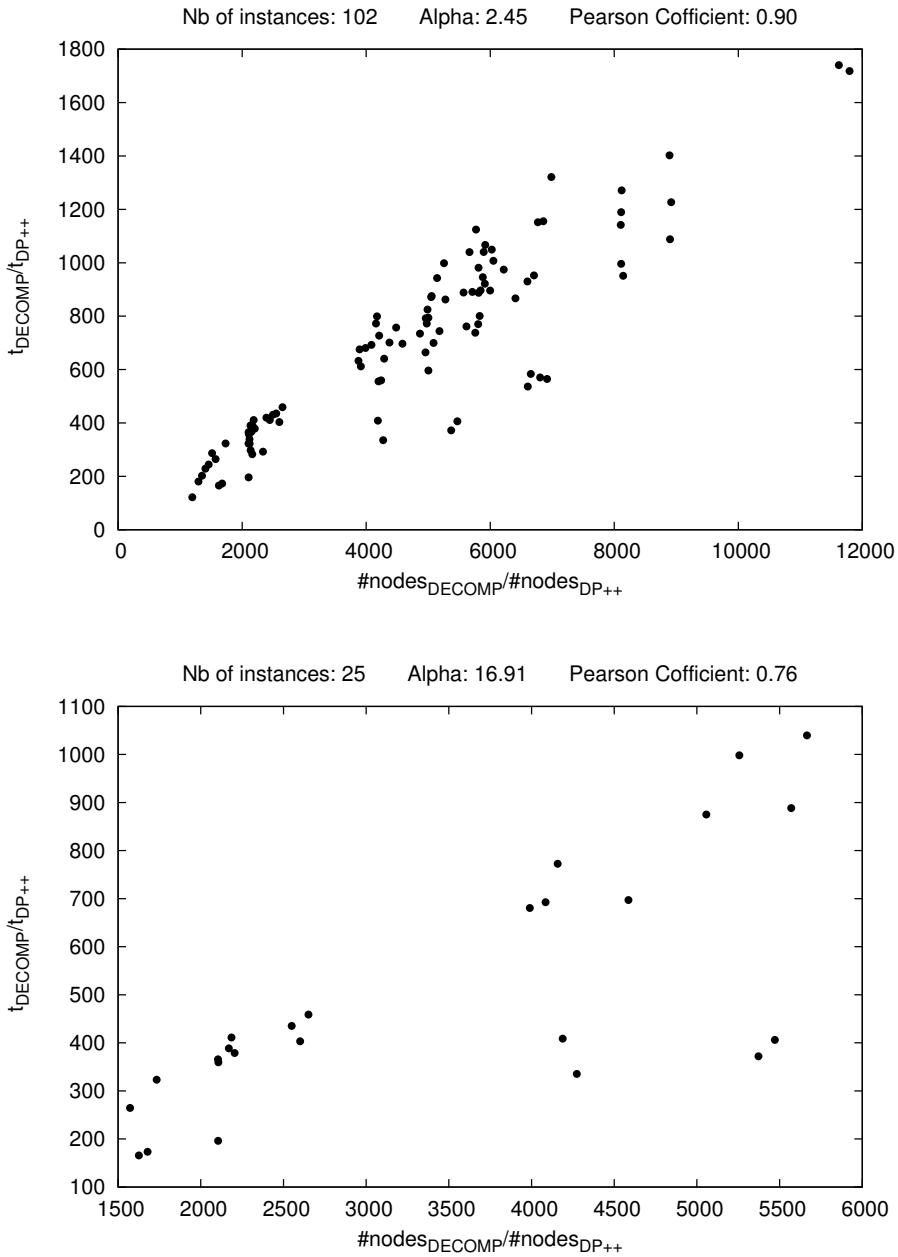


Figure 4.25: **Paucity of data effects the Pearson Correlation Coefficient.** Scatter plots corresponding to two Pearson coefficients in the Fig. SI-4.18 at 0.1% noise level. A small number of instances (bottom figure) yields a scatter plot loosely distributed, resulting in a decrease of the Pearson correlation coefficient despite the increase in alpha value.

Chapter 5

Connectivity Inference: the Unweighted Case [1]

5.1 Introduction

5.1.1 Connectivity Inference for Macro-molecular Assemblies

Macro-molecular assemblies. Building models of macro-molecular machines is a key endeavor of biophysics [Fer99], as such models not only unravel fundamental mechanisms of life [SZZ07], but also offer the possibility to monitor and to fix defaulting systems. Example of such machines are the eukaryotic initiation factors which initiate protein synthesis by the ribosome, the ribosome which performs the synthesis of a polypeptide chain encoded in a messenger RNA derived from a gene, chaperonins which help proteins to adopt their 3D structure, the proteasome which carries out the elimination of damaged or misfolded proteins, etc. These macro-molecular assemblies involve from tens to hundreds of molecules, and range in size from a few tens of Angstroms (the size of one atom) up to 100 nanometers.

But if atomic resolution models of small assemblies are typically obtained with X-ray crystallography and/or nuclear magnetic resonance, large assemblies are not, in general, amenable to such studies. Instead, their reconstruction by *data integration* requires mixing a panel of complementary experimental data [AFK+08]. In particular, information on the hierarchical structure of an assembly, namely its decomposition into sub-complexes (complexes for short in the sequel) which themselves decompose into isolated molecules (proteins or nucleic acids) can be obtained from mass spectrometry.

Mass spectrometry. Mass spectrometry (MS) is an analytical technique allowing the measurement of the mass-to-charge (m/z) ratio of molecules [SAR12], based on three devices, namely a source to produce ions from samples in solution, an analyzer separating them according to their m/z ratio, and a detector to count them. The process results in a m/z spectrum, whose deconvolution yields a mass spectrum, i.e. an histogram recording the abundance of the various complexes as a function of their mass. Considering this spectrum as raw data, two mathematical questions need to be solved. The first one, known as stoichiometry determination (SD), consists of inferring how many copies of the individual molecules are needed to account for the mass of a mode of the spectrum [BL07, ACMD14]. The second one, known as connectivity inference, aims at finding the most plausible connectivity of the molecules involved in a solution of the SD problem.

Connectivity inference. Given a macro-molecular assembly whose individual molecules (proteins or nucleic acids) are known, we aim at inferring the connectivity between these molecules. In other words, we are given the vertices of a graph, and we wish to figure out the edges it should have. To constrain the problem, we assume that the composition, in terms of individual molecules, of selected complexes of

[1] The theoretical results presented in this chapter were obtained by my co-authors, see the conference publication [AAC+13].

the assembly is known. Mathematically, this means that the vertex sets of selected *connected subgraphs* of the graph sought are known Fig. 5.1 for illustration). To see where this information comes from, recall that a given assembly can be chemically denatured i.e. split into complexes by manipulating the chemical conditions prior to ionization. In extreme conditions, complete denaturation occurs, so that the individual molecules can be identified using MS. In milder conditions, multiple overlapping complexes are generated: once the masses of the proteins are known, the list of proteins in each such complex is determined by solving the aforementioned SD problem [SR07]. In inferring the connectivity, *smallest-size networks* (i.e. graphs with as few edges as possible) are sought [ADV+07, THS+08]. Indeed, due to volume exclusion constraints, a given protein cannot contact all the remaining ones, so that the minimal connectivity assumption avoids speculating on the exact (unknown) number of contacts.

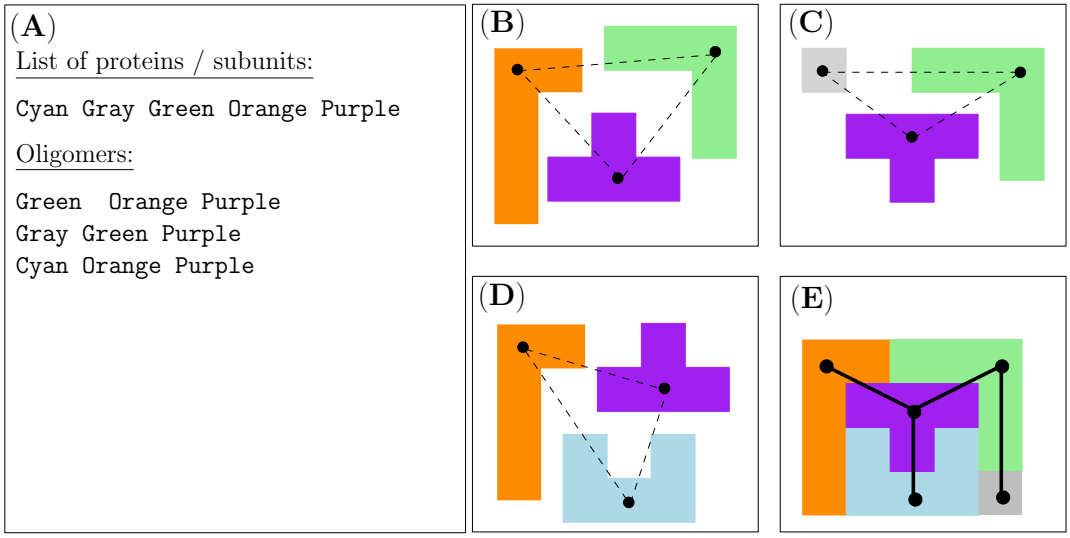


Figure 5.1: **Minimum Connectivity Inference: illustration on a fictitious system.** Given an assembly whose subunits are known but pairwise contacts are not, and for which the composition of a number of oligomers in terms of subunits is also known, the problem consists of inferring contacts between subunits. On this toy example, it is assumed that three oligomers (three trimers) are known (panels **(A, B, C)**). To connect each oligomer using as few edges as possible, two edges must be chosen, out of three possible. The *Minimum Connectivity Inference* consists of finding the overall smallest number of edges such that each each oligomer gets connected. Panel (**(D)**) shows one optimal solution with 4 edges (bold edges). Note also that these four edges from a subset of all pairwise contacts.

Mathematical Model. We refer the reader to [BM08] for basic notions of graph theory and algorithms. Let $G = (V, E)$ be a graph, where V is the set of vertices and E the set of edges. We denote $G[V']$, respectively $G[E']$, the subgraph of G induced by $V' \subseteq V$, resp. by $E' \subseteq E$.

Consider an assembly together with the list of constituting proteins, as well as a list of associated complexes. Prosaically, we associate to each protein a vertex $v \in V$ and to each complex a subset $V_i \subseteq V$. The set of all complexes is denoted $\mathcal{C} = \{V_i \mid V_i \subseteq V \text{ and } i \in I \subseteq \mathbb{N}\}$. Our goal is to infer the connectivity inside each complex of proteins. In other words, we look for contacts between proteins inside the complexes. Therefore, we need to select a set of edges, or contacts, E_i between the vertices of V_i such that the graph $G_i = (V_i, E_i)$ is connected. The Minimum Connectivity Inference problem is to find a graph $G = (V, E)$ with minimum cardinality set of edges E such that the subgraph $G[V_i]$ induced by each $V_i \in \mathcal{C}$, is connected. Formally, we state the problem as follows.

Definition. 1 (Minimum Connectivity Inference problem, MCI)**.**

Inputs: *A set V of n vertices (proteins) and a set of subsets (complexes) $\mathcal{C} = \{V_i \mid V_i \subseteq V$ and $i \in I\}$.*

Constraint: *A set E of edges is* feasible *if $G[V_i] \subseteq G = (V, E)$ is connected, for every $V_i \in \mathcal{C}$.*

Output: *A feasible set of edges (contacts) E with minimum cardinality.*

Optimality of solutions on solving MCI. The problem of connectivity inference (CI) is to find the set of edges complying with the oligomers (vertex sets). We do not know the size of this edge set apriori and do not speculate either. Using *Minimum Connectivity Inference* problem (MCI), namely the variant of CI one finds the edge set of minimal cardinality.

For each oligomer of size, s, it reuires s-1 edges to be connected (tree-connected). However, it should be noted then when edges corresponding to all the oligomers are put on the final graph, G, it may or may not have cycles. In the above Fig. 5.1, the final graph does not have cycles. The contrasting example is given in the Fig. 5.2.

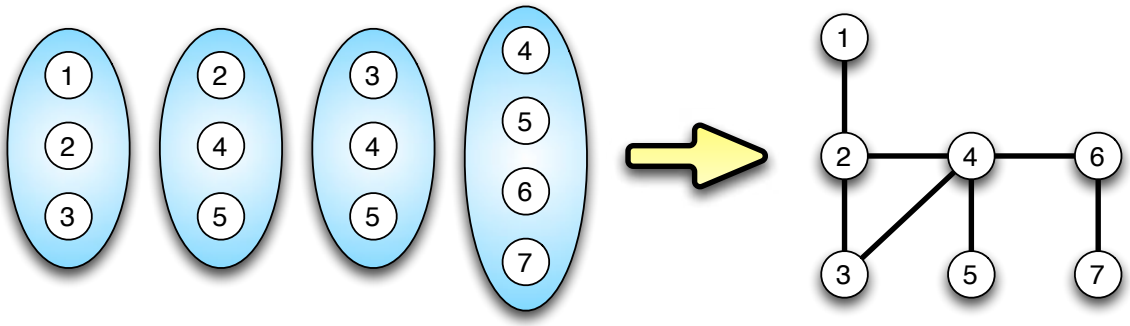


Figure 5.2: **Minimum Connectivity Inference: Cycles might arise in the optimal solutions.** One of the possible optimal solutions of minimal cardinality is shown for the system of four oligomers. Each oligomer is minimally connected in the final graph but on combining edges for all oligomers the final graph has a cycle.

Related work. The connectivity inference problem was first addressed in [THS+08] using a two-stage algorithm, called *network inference* (`NI` in the sequel). First, random graphs meeting the connectivity constraint are generated, by incrementally adding edges at random. Second, a genetic algorithm is used to reduce the number of edges, and also boost the diversity of the connectivity. Once the average size of the graphs stabilizes, the pool of graphs is analyzed to spot highly conserved edges.

From the Computer Science point of view, MCI is a network design problem in which one wants to choose a set of edges with minimum cost to connect entities (e.g., routers, antennas, etc.) subject to particular connectivity constraints. Typical examples of such constraints are that the subgraph must be k-connected, possibly with minimum degree or maximum diameter requirements (see [Rag95] for a survey). Such network design problems are generally hard to solve. To the best of our knowledge, the problem of ensuring the connectivity for different subsets of nodes has not been addressed before.

5.1.2 Outline

This paper is organized as follows. In Section 5.2, we first introduce some reduction rules to simplify the instances of MCI and we prove that the MCI problem is NP-hard. We then present a mixed integer linear programming formulation (`MILP`) formulation for the Minimum Connectivity Inference problem in Section 5.3, and an approximate greedy algorithm in Section 5.4. In Section 5.5, we validate our resolution methods using assemblies that have been investigated by mass spectrometry. We generate all optimal solutions of MCI using either MILP or the greedy algorithm and compare them with a reference set of contacts determined experimentally. The results show that our methods are effective in terms of precision

and score of the solutions, which should leverage the interpretation of protein complexes obtained by mass spectrometry.

5.2 Preliminaries and Hardness

5.2.1 Simplifying an Instance of MCI: Reduction Rules

Let $(V, \mathcal{C})$ be an instance of MCI. We denote by $(V, \mathcal{C}) \setminus u$ the instance of MCI obtained from $(V, \mathcal{C})$ by removing u from V and from all the subsets of $\mathcal{C}$ it belongs to. Moreover, we denote by $\mathrm{OPT}((V, \mathcal{C}))$ the cardinality of an optimal solution of MCI for the instance $(V, \mathcal{C})$. Let us now denote $\mathcal{C}(v) = \{V_i \mid v \in V_i\} \subseteq \mathcal{C}$, the set of complexes containing the protein $v \in V$. We observe that we can apply the following reduction rules to any instance of MCI:

Lemma. 1 (Reduction Rules)**.** *Let $(V, \mathcal{C})$ be an instance of MCI.*

1. *If $V_i \in \mathcal{C}$ is such that $|V_i| = 1$, then any feasible solution for $(V, \mathcal{C} \setminus V_i)$ is also feasible for $(V, \mathcal{C})$, and we have $OPT((V, \mathcal{C} \setminus V_i)) = OPT((V, \mathcal{C}))$;*

2. *If $\mathcal{C}(u) \subseteq \mathcal{C}(v)$, for some $u, v \in V$, then a feasible solution for $(V, \mathcal{C})$ is obtained from a feasible solution for $(V, \mathcal{C}) \setminus u$ by adding the edge uv, and we have $OPT((V, \mathcal{C})) = OPT((V, \mathcal{C}) \setminus u) + 1$.*

Proof. Statement 1 follows from the fact that V_i always induces a connected subgraph, since it contains a single vertex.

For Statement 2, observe first that given a feasible solution for $(V, \mathcal{C}) \setminus u$, we can construct a feasible solution for $(V, \mathcal{C})$ by adding the edge uv. Hence we have $\mathrm{OPT}((V, \mathcal{C})) \leq \mathrm{OPT}((V, \mathcal{C}) \setminus u) + 1$.

To prove that $\mathrm{OPT}((V, \mathcal{C}) \setminus u) \leq OPT((V, \mathcal{C})) - 1$, it suffices to show that starting from any feasible solution E for instance $(V, \mathcal{C})$, one can construct a feasible solution E^* with the same cardinality ($|E^*| = |E|$) and such that the only edge incident to u in E^* is uv. Indeed, as $\mathcal{C}(u) \subseteq \mathcal{C}(v)$, $E^* \setminus uv$ is a feasible solution for $(V, \mathcal{C}) \setminus u$.

More precisely, suppose first that $uv \notin E$. Let $P = uw_1 \ldots w_p v$ be an uv-path in G, $p \geq 1$. Such a path exists since E is a feasible solution for $(V, \mathcal{C})$ and $\mathcal{C}(u) \subseteq \mathcal{C}(v)$. So for every $V_i \in \mathcal{C}(u)$, $G[V_i]$ is a connected subgraph containing both u and v. Observe that the edge uw_1 only appears in the subgraphs $G[V_i]$, for $V_i \in \mathcal{C}(u) \cap \mathcal{C}(w_1)$. Then we claim that the set E' obtained from E by removing edge uw_1 and adding edge uv is also a feasible solution for $(V, \mathcal{C})$. In fact, the removal of uw_1 can only disconnect the subgraphs $G[V_i]$ for $V_i \in \mathcal{C}(u)$. Moreover, the subgraphs $G[V_i]$ that become disconnected after the removal of uw_1 will have exactly two connected components, one containing u and the other containing v, since there exists a path from w_1 to v. Then, the addition of edge uv reconnects the disconnected subgraphs, and we have $|E| = |E'|$. Suppose now that $uv \in E$ and that there exists an edge $uw \in E$ with $w \neq v$. By using a similar argument, we construct the set E'' of edges from E by removing the edge uw and adding the edge vw. Note that, E'' is a feasible solution for $(V, \mathcal{C})$ and $|E| = |E''|$. □

By applying Lemma 1, we conclude that we can reduce the input instances of MCI to instances where every subset V_i has at least two vertices, every vertex appears in at least two subsets V_i and V_j with $i \neq j$, and the sets $\mathcal{C}(u)$ and $\mathcal{C}(v)$ are different, for any two vertices u and v.

5.2.2 Hardness

We refer the reader to [Vaz01] for notions of computational complexity and approximation algorithms.

We establish that MCI is APX-hard, by showing a reduction from the Set Cover problem. The Set Cover problem is defined as follows:

Definition. 2 (Set Cover problem)**.**

Inputs: *a ground set $X = \{x_1, \ldots, x_m\}$, a collection $\mathcal{F} = \{X_j \subseteq X,\ j \in J\}$ and a positive integer k.*

Question: *does there exist $J' \subseteq J$ such that $\bigcup_{j \in J'} X_j = X$ and $|J'| \leq k$?*

It is well-known that the Set Cover problem is NP-complete [Kar72] and that this problem cannot be approximated in polynomial-time by a factor of $\ln n$, unless $P = NP$ [LY94, AMS06]. In order to prove our NP-completeness result, let us formally define the decision version of MCI as:

Definition. 3 (Decision version of the Connectivity Inference problem, CI)**.**

Inputs: *A set of vertices V, a collection of subsets $\mathcal{C} = \{V_i \mid V_i \subset V$ and $i \in I\}$ and a positive integer k.*

Constraint: *A set E of edges is* feasible *if $G[V_i] \subseteq G = (V, E)$ is connected, for every $V_i \in \mathcal{C}$.*

Question: *Does there exists a feasible set E of edges such that $|E| \leq k$?*

Theorem. 1. *The decision version of the* Connectivity Inference *problem is NP-complete.*

Proof. Given a set E of edges, one can check in polynomial time whether $|E| \leq k$ and each induced subgraph $G[V_i]$ is connected, for every $V_i \in \mathcal{C}$. Therefore the problem is in NP.

Let $\mathcal{I}_{SC} = (X, \mathcal{F}, k)$ be an instance of the Set Cover problem. We associate with it in polynomial time an instance $\mathcal{I}_{CI} = (V, \mathcal{C}, k')$ of the decision version of the Connectivity Inference problem as follows:

1. To each subset $X_j \in \mathcal{F}$, we associate two vertices a_j and b_j. So $|V| = 2|\mathcal{F}|$;
2. To each element $x_i \in X$, we associate the complex $V_i = \{a_j, b_j \mid x_i \in X_j, j \in J\}$;
3. We add the $2\binom{|\mathcal{F}|}{2}$ complexes of two vertices $A_{jj'} = \{a_j, a_{j'}\}$ and $B_{jj'} = \{b_j, b_{j'}\}$, where $j, j' \in J$;
4. $k' = 2\binom{|\mathcal{F}|}{2} + k$.

We show that $\mathcal{I}_{SC}$ is true if and only if $\mathcal{I}_{CI}$ is true.

First, note that any feasible solution of $\mathcal{I}_{CI}$ contains the $2\binom{|\mathcal{F}|}{2}$ edges $a_j a_{j'}$ and $b_j b_{j'}$ for all the pairs $j, j' \in J$, in order to ensure the connectivity of the complexes $A_{jj'}$ and $B_{jj'}$. We denote this set of edges by E_C. Let $J' \subseteq J$, $|J'| \leq k$ be a true solution for $\mathcal{I}_{SC}$. Let $E = E_C \cup E_k$ where $E_k = \{a_j b_j \mid j \in J'\}$, $|E_k| \leq k$. We claim that E is feasible. Indeed, thanks to the edges of E_C, the two subgraphs induced by $A_{jj'}$ and by $B_{jj'}$ are connected. For each $i \in I$, there exists a set X_{j_i} with $j_i \in J'$ and so the edges $a_{j_i} b_{j_i}$ of E_k make the subgraph induced by V_i connected.

Conversely, let E be a feasible solution for $\mathcal{I}_{CI}$. Recall that it contains the $2\binom{|\mathcal{F}|}{2}$ edges of E_C plus a set E_k of at most k edges of the form $a_j b_{j'}$. Let J' be the set of indices j such that a_j is the endpoint of one edge of E_k. Then, J' is a feasible solution for $\mathcal{I}_{SC}$. Indeed, as V_i is connected, it contains one edge $a_j b_{j'}$ of E_k with $j \in J'$, and so the element x_i belongs to X_j, which means that $\bigcup_{j \in J'} X_j = X$. □

From the reduction used in the proof of Theorem 1 and the previous results on Set Cover problem [LY94, AMS06], we conclude that MCI is APX-hard:

Corollary. 1. *There exists a constant $\mu > 0$ such that approximating MCI within a factor $1 + \mu$ is NP-hard.*

5.3 Solving the Problem to Optimality using Mixed Integer Linear Programming

Enforcing the connectivity of a subgraph $G[V_i]$ of $G = (V, E)$ can be modeled in different ways. The most effective way is to force the existence of a flow in-between any two vertices of the same subset. In the sequel, we present a mixed integer linear programming formulation (`MILP`) minimizing the number of edges needed to ensure the existence of a flow inside each complex.

To solve an instance $(V, \mathcal{C})$ of the MCI problem, we introduce one binary variable y_e for each edge $e = uv$ of the undirected complete graph on $|V|$ vertices $K_{|V|}$, to determine whether edge e is selected in the solution.

Thus, the objective function consists of minimizing the sum of the y variables, as specified by Eq. (5.1). To solve this problem, we form the directed graph $D = (V, A)$ in which each edge $e = uv$ of the complete graph $K_{|V|}$ is replaced by two directed arcs (u, v) and (v, u). The solution using MILP satisfies the following constraints:

- ▷ *Connectivity constraints.* To enforce the connectivity of each complex, we select one vertex s_i per subset $V_i \in \mathcal{C}$ as the source of a flow that must reach all other vertices in V_i using only arcs in $D[V_i]$. We introduce continuous variables $f^i_{(u,v)} \in \mathbb{R}^+$ to express the quantity of flow originating from s_i and circulating along the arc (u, v) from node u to v, with $u, v \in V_i$. Constraints (5.2), the flow conservation constraints, express that $|V_i| - 1$ units of flow are sent from s_i, and each vertex u_i collects 1 unit of flow from s_i and forwards the excess it has received from s_i to its neighbors in $D[V_i]$.

- ▷ *Capacity constraints.* We also introduce a continuous variable $x_{(u,v)} \in [0, 1]$, with $(u, v) \in A$ and $u, v \in V$, that is strictly positive if arc (u, v) carries some flow and 0 otherwise. In other words, no flow can use arc (u, v) when $x_{(u,v)} = 0$ as ensured by Constraints (5.3).

- ▷ *Symmetry constraints.* If there is some flow on arc (u, v) or (v, u) in D, then variable x is strictly positive and so the corresponding edge uv must be selected in the solution, meaning that $y_{uv} = 1$, as ensured by Constraints (5.4) and (5.5).

Denoting E_K the edges of the complete graph $K_{|V|}$, and $A^+_i(u)$ (resp. $A^-_i(u)$) the subset of arcs of $D[V_i]$ entering (resp. leaving) node u, the formulation reads as:

$$\min \quad \sum_{uv \in E_K} y_{uv} \tag{5.1}$$

$$\text{s.t.} \quad \sum_{(u,v) \in A^+_i(u)} f^i_{(u,v)} - \sum_{(w,u) \in A^-_i(u)} f^i_{(w,u)} = \begin{cases} |V_i| - 1 & \text{if } u = s_i \\ -1 & \text{if } u \neq s_i \end{cases} \qquad \forall\, u \in V_i,\ \forall\, V_i \in \mathcal{C} \tag{5.2}$$

$$f^i_{(u,v)} \leq |V_i| \cdot x_{(u,v)}, \qquad \forall\, V_i \in \mathcal{C},\ \forall\, (u, v) \in A \tag{5.3}$$

$$x_{(u,v)} \leq y_{uv}, \qquad \forall\, uv \in E_K \tag{5.4}$$

$$x_{(v,u)} \leq y_{uv}, \qquad \forall\, uv \in E_K \tag{5.5}$$

$$0 \leq x_{(u,v)} \leq 1 \qquad \forall\, uv \in E_K \tag{5.6}$$

$$0 \leq x_{(v,u)} \leq 1 \qquad \forall\, uv \in E_K \tag{5.7}$$

$$y_{uv} \in \{0, 1\} \qquad \forall\, uv \in E_K \tag{5.8}$$

Observe that this formulation can be turned into a decision formulation, by removing the objective function and adding the constraint of Eq. (5.9). If the formulation becomes infeasible, the optimal solution has more than k edges.

$$\sum_{uv \in E_K} y_{uv} \leq k \tag{5.9}$$

$$\sum_{uv \in E_\ell} y_{uv} < |E_\ell| \qquad \forall\, E_\ell \in \mathcal{S} \tag{5.10}$$

Remark 4. We can use the decision formulation to enumerate all optimal solutions (i.e. for the instance $(V, \mathcal{C}, OPT)$, where OPT is the optimal value of MCI) and report an ensemble of solutions, denoted $\mathcal{S}_{\text{MILP}}$ in the sequel. To do so, we use Constraints (5.10), where $\mathcal{S}$ is the set of solutions that have already been found. These constraints prevent finding twice a solution. We first set $\mathcal{S}$ to an optimal solution obtained after one resolution, then we add to it all newly found solutions and repeat until the problem becomes infeasible for a solution of size OPT. We finally export all optimal solutions from $\mathcal{S}$ to $\mathcal{S}_{\text{MILP}}$.

5.4 Approximate Solution based on a Greedy Algorithm

5.4.1 Design and Properties

We now propose a greedy algorithm for MCI. Starting from the empty graph $G^0 = (V, E^0 = \emptyset)$, Algorithm 2 iteratively builds a graph $G^t = (V, E^t)$, with $E^t = E^{t-1} \cup \{e^t\}$. The edge $e^t = uv$ chosen at step t aims at maximizing the reduction on the number of connected components in the induced subgraphs $G^{t-1}[V_i]$ of G^{t-1}, for $V_i \in \mathcal{C}(u) \cap \mathcal{C}(v)$. More formally, at step t, we choose an edge e^t maximizing $m_t(e^t = uv)$ among all pairs $u, v \in V$, with

$$m_t(e^t = uv) = |\{i \mid V_i \ni u, V_i \ni v, \text{and } u \text{ and } v \text{ are in distinct connected components of } G^{t-1}[V_i]\}|.$$

The quantity $m_t(e^t = uv)$ is called the *priority* of the edge e^t.

Algorithm 2 Greedy algorithm for MCI

Require: $V = \{v_1, \ldots, v_n\}$ and $C = \{V_i \mid V_i \subseteq V \text{ and } i \in I\}$.
Ensure: A set E of edges such that $G[V_i] \subseteq G = (V, E)$ is connected, for every $i \in I$.

```
t := 1, E^0 := ∅, G^0 := G(V, E^0)
while there exists a disconnected graph G^{t-1}[V_i], for some i ∈ I do
    Find edge e^t maximizing the priority m_t(e^t)
    E^t := E^{t-1} ∪ {e^t} and t := t + 1
return E_{t-1}
```

When comparing Greedy with the MILP formulation, one can easily find example where Greedy is outperformed by algorithm MILP (see Fig. 5.3 for an example). In the example of Fig. 5.3, the optimal solution contains 6 edges (uv, uz, vw, vy, wz, xz). It is optimal as any feasible set of edges should contain at least 2 edges in each complex 4, 7, 10 (which contains no pair of vertices in common). Let us now apply algorithm Greedy on this example. There is an unique edge vz with priority 6 (the maximum value). So we have to choose it at step 1. For step 2 we can choose any of the edges with priority 5 : uv, uz, vw, and wz. Suppose we choose uv; then at step 3 we can choose one of the two edges which still have priority 5 : vw and wz. Then we need to choose at steps 4 and 5 both edges vy and xz. Ii remains to choose two edges among the vertices u, w, z. The solution obtained has 7 edges and is not optimal as there exists an optimal solution with 6 edges. However the next result shows that it is a $2(\log_2 M)$-approximation.

1 :	u	v	w	x	z
2 :	u	v	w	y	
3 :	u	v	x	z	
4 :	u	v	y		
5 :	u	v	y	z	
6 :	u	w	z		
7 :	u	x	z		
8 :	v	w	x	z	
9 :	v	w	y	z	
10 :	v	w	z		

(a) Instance description: ten complexes defined from 6 vertices.

Algorithm	# sols	sol. size	Selected edges
MILP	1	6	uv, uz, vw, vy, wz, xz
Greedy	9	7	1: vz, uv, vw, vy, xz, uz, uw
			2: vz, uv, vw, vy, xz, uz, wz
			3: vz, uv, wz, vy, xz, uw, ux
			4: vz, uv, wz, vy, xz, uw, uz
			5: vz, uv, wz, vy, xz, uz, wy
			6: vz, uz, vw, vy, xz, uw, uy
			7: vz, uz, vw, vy, xz, uy, wz
			8: vz, uz, wz, vy, xz, uy, uw
			9: vz, uz, wz, vy, xz, uy, wy

(b) Solutions and statistics. MILP yields one solution with 6 edges. Greedy yields 9 solutions with 7 edges. Selected edges in each solution are reported in the order of selection by Greedy.

Figure 5.3: **Sub-optimality of `Greedy` over `MILP`: example MCI problem.** On this example, `MILP` and `Greedy` respectively yield solutions with 6 and 7 edges (or contacts). When using MILP to report solution with 7 edges, we get 16 solutions.

Proposition. 1. *Let $M = \max_{u,v \in V} |\mathcal{C}(u) \cap \mathcal{C}(v)|$, Algorithm 2 is a $2(\log_2 M)$-approximation algorithm for MCI.*

Proof. We divide the steps of the algorithm into $\log_2 M$ phases. During each phase p, the value of $m_t(e^t)$ remains in the interval $[a_{p+1}, a_p]$ with $a_1 = M$ and $a_{p+1} = a_p/2$. Let Λ_p be the number of selected edges during phase p. We observe that the size of the output of Algorithm 2 is $SOL = \sum_{p=1}^{\log_2 M} \Lambda_p$. Let also δ_p be the number of components of the graphs $G^t[V_i]$ that have been connected during that phase. We have:

$$a_{p+1}\Lambda_p \leq \delta_p \leq a_p \Lambda_p \tag{5.11a}$$

Since during phase p we have reduced the number of components by δ_p, we know that the remaining number of components to connect at the beginning of the phase was at least δ_p. Furthermore, the maximum value $m_t(e)$ of an edge during phase p is upper bounded by a_p. So to connect these remaining components, we need at least δ_p/a_p edges. Hence we have

$$OPT \geq \frac{\delta_p}{a_p} \tag{5.11b}$$

Using Eq. 5.11a, we obtain that $OPT \geq \frac{a_{p+1}}{a_p}\Lambda_p = 2\Lambda_p$. Now, summing over all phases, we obtain

$$SOL = \sum_{p=1}^{\log_2 M} \Lambda_p \quad \leq \quad \sum_{p=1}^{\log_2 M} 2 \cdot OPT = 2 \cdot OPT \cdot \log_2 M \tag{5.11c}$$

□

Proposition. 2. *When $\max_{v \in V} |\mathcal{C}(v)| = 2$, Algorithm 2 always returns an optimal solution.*

Proof. By Lemma 1, we may assume that $m_t(e) \leq 1$, for every $e \in E$ and for every t. Consequently, when an edge e^t is chosen, it will be useful to connect only one complex. Thus, all the edges that are chosen by the algorithm are necessary (in the sense that one edge would be necessary to connect such complex) and the solution is optimal. □

5.4.2 Implementation

In the following, we sketch an implementation of Algorithm 2, denoted `Greedy` in the sequel, which does not scan every candidate edge in E^t to find the (or a) best one, but instead maintains the priorities of all candidate edges.

Consider the following data structures:

- a priority queue Q associating to each candidate edge e its priority defined by $m_t(e)$. Note that the initial priority is given by $m_0(e = uv) = |\mathcal{C}(u) \cap \mathcal{C}(v)|$.
- a union-find data structure UF_i used to maintain the connected components of the induced graph $G^t[V_i]$. We assume in particular the existence of a function Find_vertices() such that UF_i.Find_vertices(u) returns the vertices of the connected component of the graph $G^t[V_i]$ containing the vertex u.

Upon popping the edge $e^t = (u, v)$ from Q, the following updates take place:

Update of the priority queue Q. For each complex V_i such that e^t triggers a merge between two connected components of $G^t[V_i]$, consider the two sets of vertices associated to these components, namely $K_{i,u} = UF_i$.Find_vertices(u) and $K_{i,v} = UF_i$.Find_vertices(v). The priority of all edges in the set $K_{i,u} \times K_{i,v} \setminus \{e^t\}$ is decreased by one unit.

Update of the union-find data structures. For each complex V_i such that e^t triggers a merge between two connected components of $G^{t-1}[V_i]$, the union operation UF_i.Union(UF_i.Find(u), UF_i.Find(v)) is performed.

It should be noticed that up to the logarithmic factor involved in the maintenance of Q, and up to the factor involving the inverse of Ackermann's function to run the union and find operations [Tar83], the update complexity is output sensitive in the number of candidate edges affected in $K_{i,u} \times K_{i,v}$.

Remark 5. Algorithm 2 can be modified to report an ensemble of solutions, denoted $\mathcal{S}_{\text{Greedy}}$ in the sequel, as follows: whenever several edges have the same priority, one forks one exploration tree for each choice of such an edge. To avoid rediscovering the same solution several times, solutions are stored in a dictionary using the lexicographic ordering on edges. The results reported in Section 5.5 were generated with `Greedy` incorporating this modification.

5.5 Experimental Results

5.5.1 Assemblies of Interest and Reference Contacts

Assemblies. We selected three assemblies investigated by mass spectrometry (MS), for which we also found reference contacts (i.e. connecting edges) between pairs of constituting proteins. These three assemblies are:
▷ *Yeast Exosome.* The exosome is a 3'- 5' exonuclease complex involved in RNA processing and degradation. The yeast exosome is composed of 10 different protein types with unit stoichiometry.

A total of 19 distinct complexes were generated from the assembly using tandem mass spectrometry (MS/MS) and subdenaturing concentration of organic solvents [HDT+06] (see supplemental Section 8.1).

▷ *Yeast 19S Proteasome lid.* Proteasomes are protein assemblies involved in the elimination of damaged or misfolded proteins, and the degradation of short-lived regulatory proteins. The most common form of proteasome is the 26S, which involves two filtering caps (the 19S), each cap involving a peripheral lid, composed of 9 distinct protein types each with unit stoichiometry.

Series of overlapping complexes were formed by MS, MS/MS and cross-linking using BS3. In total, 14 complexes were obtained out of which 8 came from MS, MS/MS and 6 came from cross-linking experiments [STA+06] (supplemental Section 8.1).

▷ *Eukaryotic Translation factor eIF3.* Eukaryotic initiation factors (eIF) are proteins involved in the initiation phase of the eukaryotic translation. They form a complex with the 40S ribosomal subunit, initiating the ribosomal scanning of mRNA. Among them, human eIF3 consists of 13 different protein types each with unit stoichiometry. The eIF3 complex in this text refers to the human eIF3 unless otherwise stated.

A total of 27 complexes were generated from the assembly by manipulating the ionic strength of the solution and using MS/MS [ZSF+08] (supplemental Section 8.1). The subunit eIF3j is unstable and since none of the 27 subcomplexes comprises this subunit we exclude it from the list of protein types, leaving behind 12 protein types.

Reference contacts. A criterion in selecting complexes has been the presence of reference contacts. These contacts, determined experimentally, are meant to check whether the edges computed by our methods are *true positives* or *false positives*. (We say that an edge computed by `MILP` or `Greedy` is a *true positive* if this edge was also determined experimentally. Otherwise we say that it is a *false positive*.) We classified these experimental contacts as a function of their reliability, into three individual categories (supplemental Section 8.2.1). The first category consists of crystal contacts (set C_{Xtal}), i.e. contacts observed in a high resolution crystal structure giving access to the atomic coordinates with an accuracy which is about the size of one atom. The availability of such a crystal structure is ideal, since then, all contacts between subunits can be determined unambiguously from the relative atomic positions [LC10]. The second category consists of the contacts (set C_{Cryo}) obtained upon reconstructing a model of the assembly using cryo-electron microscopy maps. The sample is made to undergo cryo-fixation to keep it in near native conditions. Using reconstruction software, a structure of intermediate resolution is obtained, from which pairwise contacts are inferred. The third category (set C_{XL}) consists of contacts obtained by cross-linking, an experimental technique allowing one to infer the relative proximity of certain amino-acids in proteins. However, because two proteins can be nearby (at say 15Å) without touching, such contacts are not completely reliable. The fourth category

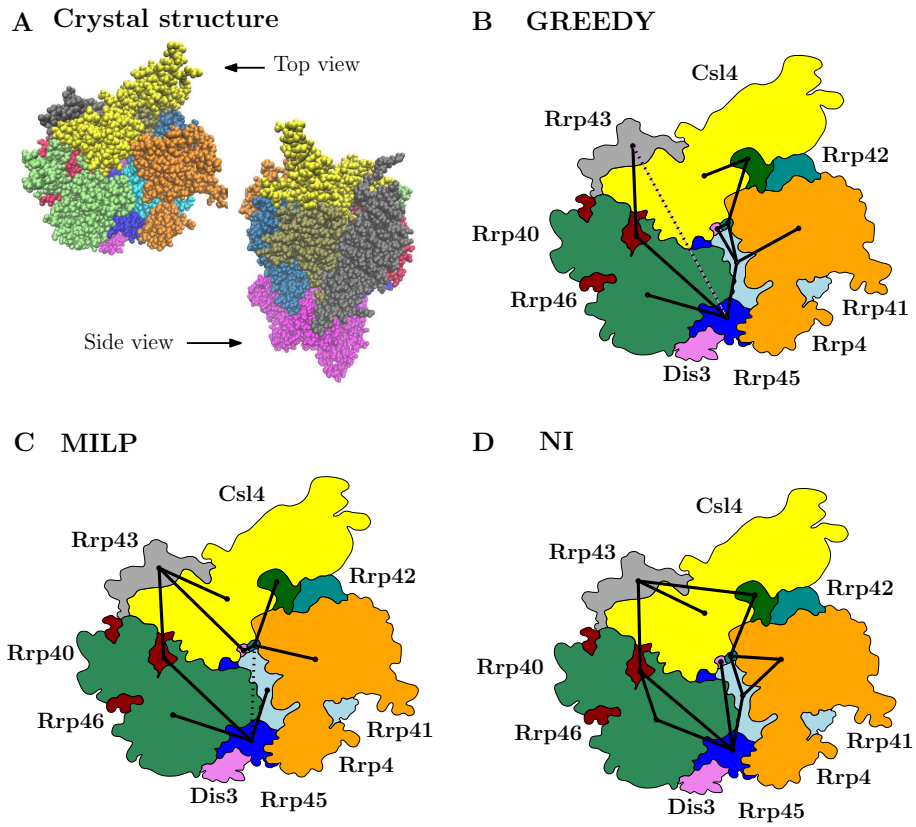


Figure 5.4: **Yeast Exosome: contacts computed by the algorithms. (A)** Top and side view of the crystal structure [MBC13, PDB 4IFD]. **(B,C,D)** Structure decorated with one edge per contact. the dash style reads as follows: *bold*: contacts in $S \cap \mathrm{C}_{\mathrm{Xtal}}$; *dotted*: contacts in S but not in $\mathrm{C}_{\mathrm{Xtal}}$; Note that a long edge i.e. an edge between two subunits that appear distant on the top view of the assembly corresponds to a contact of these subunits located further down along the vertical direction. Also, note that part of the subunits Dis3 and Rrp42 are visible in the middle of the assembly and are trapped in between Csl4, Rrp40, Rrp41. The contact node therefore is placed there for convenience. The subunit Mtr3 is not annotated and is sandwiched between Csl4, Rrp42 and Rrp4.

consists of miscellaneous dimers (set $\mathrm{C}_{\mathrm{Dim}}$) obtained by various biophysical experiments, which also feature false positives.

In presence of a crystal structure, we define the reference set of contacts as

$$\mathrm{E}_{\mathrm{Ref}} = \mathrm{C}_{\mathrm{Xtal}}. \tag{5.12}$$

In the absence of crystal structure, we define the reference contacts as

$$\mathrm{E}_{\mathrm{Ref}} = \mathrm{C}_{\mathrm{Cryo}} \cup \mathrm{C}_{\mathrm{Dim}} \cup \mathrm{C}_{\mathrm{XL}}. \tag{5.13}$$

The exosome is covered by the former case (supp. Table 8.1). The proteasome lid is covered by the latter case (supp. Table 8.2). Finally, eIF3 is also covered by the latter case, with an empty set C_{XL} (supp. Table 8.3).

Connectivity inference: illustration. Before analyzing the performances of our methods, we illustrate the connectivity inference problem using the results of the network inference algorithm `NI` [THS+08], whose set of contacts is denoted E_{NI}, on the yeast exosome (Fig. 5.4), and on the yeast 19S proteasome lid (Fig.

Assembly	#types	E_{Ref}	$\|E_{Ref}\|$	$\|E_{NI}\|$	$P_{NI;E_{Ref}}(E_{NI})$
Exosome	10	C_{Xtal}	26	12	12
19S Lid	9	$C_{Cryo} \cup C_{Dim} \cup C_{XL}$	19	9 (NC^*)	8
eIF3	12	$C_{Cryo} \cup C_{Dim} \cup C_{XL}$	17	17**	14

Table 5.1: **Size and precision of solutions for `NI`.** First section of the table: assembly, number of protein types, and size of the reference set E_{Ref}; second: size and precision for the solution returned by the algorithms `NI` [THS+08] NB: **The assignment of contacts was done manually [ZSF+08]; NC^*: assembly not connected

5.5). For that system, a total of 26 contacts should ideally be retrieved (supplemental Table 8.1), yet, the algorithm reports twelve of them:

(Csl4, Rrp43)	(Dis3, Rrp45)	(Mtr3, Rrp42)	(Mtr3, Rrp43)	(Rrp4, Rrp41)
(Rrp4, Rrp42)	(Rrp40, Rrp45)	(Rrp40, Rrp46)	(Rrp41, Rrp42)	(Rrp41, Rrp45)
(Rrp43, Rrp46)	(Rrp45, Rrp46)			

All these contacts are true positives. As we shall see, the assessment of `MILP` and `Greedy` is more challenging, since an ensemble of solutions rather than a unique solution is reported.

5.5.2 Results

Number of Solutions

Methods. The ensembles of solutions reported by `MILP` and `Greedy` are respectively denoted $\mathcal{S}_{MILP}$ and $\mathcal{S}_{Greedy}$. We first inspect the size of these ensembles, which hint at the variety of possible connections between proteins, while respecting the connectivity constraints imposed by complexes.

Results. The size of solutions sets reported by the algorithms are displayed in Tables 5.1, 5.2, and 5.3. For the exosome, `MILP` and `Greedy` respectively find 1644 and 756 solutions. For the proteasome lid, the corresponding figures are 324 and 324. Finally, for the translation factor eIF3, one gets 180 and 108 possibilities. These numbers backup the importance of enumerating solutions, as opposed to singling out a single solution, and naturally call for the inspection of the contacts involved, to check whether these are true or false positives.

Parsimony and Precision of Solutions

Methods. We define the *precision* of an individual solution as the number of contacts from the reference set E_{Ref} it contains. Since selected contacts can be missed by an individual solution, we do the same for an ensemble of solutions, by comparing the union of the contacts associated with all individual solutions in this ensemble against the reference contacts.

More formally, consider an ensemble of solutions $\mathcal{S}_A = \{E_A\}$ reported by algorithm A ($\mathcal{S}_{MILP}$ reported by `MILP` or $\mathcal{S}_{Greedy}$ reported by `Greedy`). The *size of a solution* $E_A \in \mathcal{S}_A$, denoted $|E_A|$, is its number of edges. The *precision* of the solution E_A w.r.t. a reference set of contacts E_{Ref} is defined as the size of the intersection, i.e. $P_{A;E_{Ref}}(E_A) = |E_A \cap E_{Ref}|$. The precision is maximum if $E_A \subseteq E_{Ref}$, in which case no predicted contact is a false positive. The notion of precision makes sense if the reference contacts are exhaustive, which is the case for the yeast exosome (since a crystal structure is known) and for the yeast proteasome lid (cryo-EM reconstruction and exhaustive list of cross-links). We summarize the precision of the ensemble of solutions $\mathcal{S}_A$, denoted $P_{A;E_{Ref}}(\mathcal{S}_A)$, by the statistical summary (minimum, median, maximum) of the precisions of the solutions in $\mathcal{S}_A$. That is, denoting Q a set of precisions (one for each solution from an ensemble of solutions), we report the triple

$$\mathrm{MMM}(Q) = (\min(Q), \mathrm{median}(Q), \max(Q)). \tag{5.14}$$

Complex	#types	E_{Ref}	$\|E_{Ref}\|$	$\|E_{MILP}\|$	$\|\mathcal{S}_{MILP}\|$	$P_{MILP;E_{Ref}}(\mathcal{S}_{MILP})$	$\|\mathcal{S}^{cons.}_{MILP}\|$	$P_{MILP;E_{Ref}}(\mathcal{S}^{cons.}_{MILP})$
Exosome	10	C_{Xtal}	26	10	1644	(7, 9, 10)	12	(8, 9, 10)
19*S Lid*	9	$C_{Cryo} \cup C_{Dim} \cup C_{XL}$	19	10	324	(7, 8, 10)	18	(8, 9, 10)
eIF3	12	$C_{Cryo} \cup C_{Dim} \cup C_{XL}$	17	13	180	(8, 10, 12)	36	(9, 10, 11)

Table 5.2: **Size and precision of solutions for `MILP`.** First section of the table: assembly, number of protein types, and size of the reference set E_{Ref}; second and third sections: size and precision of algorithm `MILP`, for the whole set of optimal solutions $\mathcal{S}_{MILP}$, and for consensus solutions $\mathcal{S}^{cons.}_{MILP}$.

Complex	#types	E_{Ref}	$\|E_{Ref}\|$	$\|E_G\|$	$\|\mathcal{S}_{Greedy}\|$	$P_{Greedy;E_{Ref}}(\mathcal{S}_{Greedy})$	$\|\mathcal{S}^{cons.}_{Greedy}\|$	$P_{Greedy;E_{Ref}}(\mathcal{S}^{cons.}_{Greedy})$
Exosome	10	C_{Xtal}	26	10	756	(7, 9, 10)	756	(7, 9, 10)
19*S Lid*	9	$C_{Cryo} \cup C_{Dim} \cup C_{XL}$	19	10	324	(7, 8, 10)	18	(8, 9, 10)
eIF3	12	$C_{Cryo} \cup C_{Dim} \cup C_{XL}$	17	13	108	(9, 10, 12)	36	(9, 10, 11)

Table 5.3: **Size and precision of solutions for `Greedy`.** First section of the table: assembly, number of protein types, and size of the reference set E_{Ref}; second and third sections: size and precision of algorithm `Greedy`, for the whole set of optimal solutions $\mathcal{S}_{Greedy}$, and for consensus solutions $\mathcal{S}^{cons.}_{Greedy}$.

Results. It is first observed that on the three systems, `MILP` and `Greedy` are more parsimonious than `NI` (Tables 5.1, 5.2, and 5.3). For example, on the yeast exosome, 10 edges are used instead of 12, while this number drops from 17 to 13 for eIF3. The precision is excellent ($\geq 80\%$) for the three algorithms on the two systems where the reference set of contacts is exhaustive (yeast exosome and yeast proteasome lid).

Overall, a striking observation is that `MILP` and `Greedy` show identical performances in terms of solution size, an observation to be put back in the context of the approximation factor proved in Proposition 1. While finding example where `Greedy` is outperformed by algorithm `MILP` is easy (Fig. 5.3), understanding whether the approximation factor is tight remains an open problem.

Running time

To compare the running times of `MILP` and `Greedy`, above instances are run on 2.4 GHz Intel Core(TM)2 Duo machine equipped with 4Go of RAM. Results for the performance on the biological assemblies are shown in Table 5.4.

Protein Assembly	#Complexes	t_{MILP}	t_{Greedy}
Yeast Exosome	19	$\sim$ 1600s	$\sim$ 1s
Yeast Exosome (w/o Csl4)	15	$\sim$ 43s	$\sim$ 0.25s
Yeast Protesome 19S lid	14	$\sim$ 39s	$\sim$ 9s
Human eIF3	27	$\sim$ 24s	$\sim$ 1.3s

Table 5.4: **Running time comparison for `MILP` and `Greedy`**

We observe from the Table 5.4 that the algorithm `Greedy` outperforms the algorithm `MILP` on all the instances. However it is to be noted that the exact solution by `MILP` only takes handful of seconds to few minutes.

Scores, Solutions and Consensus Solutions

Methods. Upon running `MILP` or `Greedy`, a given contact typically appears in several individual solutions. We therefore compute its *score* i.e. the number of solutions containing it. We identify similarly *consensus solutions*, namely solutions maximizing the sum of the scores of their contacts. The contacts found in the consensus solutions are called *consensus contacts*. Prosaically, the consensus contacts define the *backbone* of the assembly, or in network terms, the *highway* connecting the individual nodes corresponding to the proteins.

More formally, let A be one of our resolution methods (`MILP` or `Greedy`) and let $\mathcal{S}_{\mathrm{A}}$ be the associated solution set. The *score of a contact* is the number of solutions from $\mathcal{S}_{\mathrm{A}}$ containing that contact, and the *signed contact score* is the contact score multiplied by ± 1 depending on whether this contact is a true or false positive w.r.t. $\mathrm{E}_{\mathrm{Ref}}$. The *score of a solution* $\mathrm{E}_{\mathrm{A}} \in \mathcal{S}_{\mathrm{A}}$ is the sum of the scores of its contacts. Finally, a *consensus solution* is a solution achieving the maximum score over $\mathcal{S}_{\mathrm{A}}$, the set of all such solutions being denoted $\mathcal{S}_{\mathrm{A}}^{cons.}$. Note that the score of a solution is meant to single out the consensus solutions from a solution set $\mathcal{S}_{\mathrm{A}}$, while the signed score is meant to assess the solutions in $\mathcal{S}_{\mathrm{A}}$ w.r.t. a reference set $\mathrm{E}_{\mathrm{Ref}}$.

Finally, we shall also compare ensembles (of solutions, or proteins). To this end, given two sets I and J, we compute the size of the intersection and of the symmetric difference:

$$I \Delta_s J = (|I \backslash J|, |I \cap J|, |I \backslash J)|). \tag{5.15}$$

Results. We first inspect scores.

▷ *Exosome.* Two facts emerge (Figs. 5.6(a), 5.6(b) and 5.7(a), 5.7(b)). First, four ubiquitous contacts are observed by `MILP`, while `Greedy` outputs an additional ubiquitous contact whose score matches the second highest score among the contacts outputted by `MILP`. The remaining contacts vary in their counts in the solution sets $\mathcal{S}_{\mathrm{MILP}}$ and $\mathcal{S}_{\mathrm{Greedy}}$. Second, there are few false positive overall. An interesting case is (Rrp42, Rrp45), which has the 7th highest count. The two polypeptide chains Rrp42 and Rrp45 are found in 14 out of 19 complexes used as input, accompanied *in all cases* by Rrp41. That is, these three chains behave like a *rigid body.*

▷ *Yeast Proteasome Lid.* The distribution of signed contact scores for the yeast proteasome lid broadly follows the similar behavior as the yeast exosome (Figs. 5.6(c), 5.6(d) and 5.7(c), 5.7(d)).

▷ *eIF3.* As for eIF3 (Figs. 5.6(e), 5.6(f) and 5.7(e), 5.7(f)), numerous false positive are observed, a fact related to the paucity of the reference contact set $\mathrm{E}_{\mathrm{Ref}}$.

We next examine consensus solutions, namely the highest scoring solutions defined from the previous scores (Tables 5.2 and 5.3, and Figs. 5.6(b), 5.6(d), 5.6(f) and 5.7(b), 5.7(d), 5.7(f)).

▷ *Yeast exosome.* Algorithms `MILP` and `Greedy`[2] respectively generate 1644 and 756 solutions: 756 are common, and the remaining ones pertain to `MILP`. Likewise, the two methods respectively report 12 and 756 consensus solutions, and one has $\mathcal{S}_{\mathrm{MILP}}^{cons.} \Delta_s \mathcal{S}_{Greedy}^{cons.} = (0, 12, 744)$. Remarkably, all solutions generated by `Greedy` are consensus solutions, i.e. $\mathcal{S}_{\mathrm{Greedy}} = \mathcal{S}_{Greedy}^{cons.}$. In moving from $\mathcal{S}_{\mathrm{MILP}}$ to $\mathcal{S}_{\mathrm{MILP}}^{cons.}$, the precision increases from $(7, 9, 10)$ to $(8, 9, 10)$. In fact, for this system, both methods have comparable performances.

▷ *Yeast proteasome lid.* Algorithms `MILP` and `Greedy` generate identical sets of solutions, 18 out of 324 being consensus solutions. In moving from all solution to consensus solutions, the precision increases from $(7, 8, 10)$ to $(8, 9, 10)$.

▷ *eIF3.* In this case, similar to the case of yeast exosome, `MILP` generated almost twice as many solutions as `Greedy`, respectively, 180 and 108, with $\mathcal{S}_{\mathrm{MILP}} \Delta_s \mathcal{S}_{\mathrm{Greedy}} = (72, 108, 0)$. However, both methods have identical set of consensus solutions, of size 36.

[2] `Greedy` incorporates forks, as explained in Remark 5.

Proteins and their Neighborhoods

Methods. For a protein p, the ideal situation is faced when all the proteins in contact with p are found. These proteins are plainly called the *neighbors* of p in the sequel. Given a set of contacts E, the neighbors of p with respect to E are denoted as follows:

$$N(p, E) = \{q \,|\, pq \in E\} \tag{5.16}$$

Using the set of reference contacts $E = \mathrm{E_{Ref}}$, one gets the reference neighbors $N(p, \mathrm{E_{Ref}})$ of p. Likewise, using the contacts of a solution $\mathrm{E_A} \in \mathcal{S}_\mathrm{A}$ reported by an algorithm A (`MILP` or `Greedy`) yields the neighbors $N(p, \mathrm{E_A})$. To measure the agreement of these two sets of neighbors, we therefore compute the triplet of Eq. (5.15) between the computed neighbors $I = N(p, \mathrm{E_A})$ and the reference neighbors $J = N(p, \mathrm{E_{Ref}})$.

An equivalent analysis can be carried out on an ensemble of solutions. To generalize the triplet of Eq. (5.15), consider a collection of sets $\mathcal{I}$ (practically: $\mathcal{I} = \{N(p, \mathrm{E_A}) \mid \mathrm{E_A} \in \mathcal{S}_\mathrm{A}\}$) and a particular set J (practically: $N(p, \mathrm{E_{Ref}})$). We define a statistical summary for the intersection and the components of the symmetric differences:

$$\Delta_s^{left}(\mathcal{I}, J) = \mathrm{MMM}(\{|I \backslash J|, I \in \mathcal{I}\}) \tag{5.17}$$

$$\Delta_s^{center}(\mathcal{I}, J) = \mathrm{MMM}(\{|I \cap J|, I \in \mathcal{I}\}) \tag{5.18}$$

$$\Delta_s^{right}(\mathcal{I}, J) = \mathrm{MMM}(\{|J \backslash I|, I \in \mathcal{I}\}) \tag{5.19}$$

Given that the previous analysis focuses on individual solutions, we also aggregate the neighborhoods of all solutions for a given protein, and compare it to the reference one. That is, we compute

$$N(p, \mathcal{S}_\mathrm{A}) \Delta_s N(p, \mathrm{E_{Ref}}) \equiv \left(\bigcup_{\mathrm{E_A} \in \mathcal{S}_\mathrm{A}} N(p, \mathrm{E_A}) \right) \Delta_s N(p, \mathrm{E_{Ref}}) \tag{5.20}$$

Results. The neighborhood analysis of each proteins was performed for the solution set $\mathcal{S}_\mathrm{MILP}$ using two biological complexes for which the set of experimentally validated contacts are exhaustive, i.e. yeast exosome and yeast proteasome lid. The degree of a protein refers to the number of its neighbors, either in the assembly or in the computed solution.

▷ *Yeast Exosome.* Consider first the degree of a protein across $\mathcal{S}_\mathrm{MILP}$. Keeping in mind that this is an optimal size solution set for the given input set of complexes, the degree in 7/10 proteins is upper bounded by 3 (supp. Table 5.5). For the remaining three proteins, degrees up to 6 and 8 are observed (supp. Table 5.5).

Consider now the neighborhoods on a protein across $\mathcal{S}_\mathrm{MILP}$ (supp. Table 5.6). For all the proteins, the median value of Δ_s^{left} is either 0 or 1, indicating that the false positive count of neighborhood detection is significantly low. On the other hand, the relatively large values of Δ_s^{right}, with median in the range 3..5, show that a unique solution falls short from providing a complete account of the reference neighborhood, an observation related to the fact that minimal connectivity solutions are sought.

To complement this latter observation, we now focus on the the assessment based on the union of neighborhoods, as provided by Eq. (5.20) (supp. Table 5.7). Comparing the reference degree of a protein against the size of the intersection of Eq. (5.18) reveals that in most cases, all the neighbors in the crystal structure are found in at least one solution in $\mathcal{S}_\mathrm{MILP}$. Yet, the entries of the symmetric difference raise two concerns, which we discuss on the extreme cases.

- Rrp43: three extra contacts are observed in the solution set with Rrp40, Rrp41 and Rrp42.
- Mtr3: the crystal contacts with Rrp4, Rrp45, and Rrp46 are missing in the solution. These three contacts are characterized by weak interfaces respectively involving 19, 24 and 54 atoms (supp. Table 8.1). Interestingly, these proteins are seen in complexes, but only large ones (the smallest contains five sub-units). That is, no complex provides a strong constraint (dimer or trimer) that may force `MILP` to include it in a solution set.

▷ *Yeast Proteasome Lid.* The degree for 6 proteins out of 9 is upper bounded by 4, whereas, degrees of 5 and 6 are observed for the three other proteins in handful number of solutions (supp. Table 5.8). The results for neighborhood detection essentially follows that of yeast exosome, i.e. low false detection and high accuracy (supp. Tables 5.9, 5.10).

Overall, the very few false positive contacts already noticed while discussing the contacts obtained as a whole are well distributed across the individual proteins. It should also be stressed that the discrepancy observed in terms of degrees owes to the fact that the solutions reported by `MILP` have optimal size, while the contacts contained in the reference contacts sets used for the assessment are exhaustive.

▷ *Human eIF3.* The degree in 9 proteins out of 12 is upper bounded by 3, the remaining cases corresponding to degree 4 (one case) and degree 5 (two cases) (supp. Table 5.11). In the union of solutions, we see series of false positives. Closer inspection shows that most of these false positives involve the interactions of the subunits eIF3i and eIF3 with the octamer core [3] of eIF3 (see Fig. 5.6(e)). The occurrence of these contacts in about 1/3 of solutions reported by `MILP` is due to the number of oligomers comprising these subunits - 4 oligomers have (c,g) (e,g) (c,i) and (e, i) and 2 oligomers of size 5 have each (a,i) (a,g). However, as discussed in Section 5.5.1, the reference contacts are not completely reliable in this case, and a high resolution crystal structure would be needed to sharpen the conclusions.

5.6 Conclusion and Outlook

A key endeavor of biophysics, for macro-molecular systems involving up to hundreds of molecules, is the determination of the pairwise contacts between these constituting molecules. The corresponding problem, known as connectivity inference, is central in mass-spectrometry based studies, which over the past five years, has proved crucial to investigate large assemblies. In this context, this paper presents a thorough study of the problem, encompassing its hardness, a greedy strategy, and a mixed integer programming model. Application-wise, the key advantage of our methods w.r.t. the algorithm *network inference* developed in biophysics, is that we fully master all optimal solutions instead of a random collection of solutions which are not qualified w.r.t. the optimum. As shown by careful experiments on three assemblies recently scrutinized by other bio-physical experiments (yeast exosome, yeast proteasome lid, eIF3), our predictions are in excellent agreement with the experimental contacts. We therefore believe that our methods should leverage the interpretation of protein complexes obtained by mass spectrometry, a research vein currently undergoing major developments.

From a theoretical standpoint, a number of challenging problems deserve further work. The first one is to determine whether the approximation factor of the greedy algorithm developed in this paper is tight or not. The second one is to understand the solution space as a function of the number of input vertex sets and the structure of the unknown underlying graph. This problem is also related to the (output-sensitive) enumeration of connected subgraphs of a given graph. The third challenge is concerned with the generalization where the stoichiometry (the number of instances) of the proteins involved is more than one. In that case, complications arise since the connectivity information associated with the vertex sets of the connected subgraphs is related to protein types, while the connectivity sought is between protein instances. This extension would allow processing cases such as the nuclear pore complex, the biggest assembly known to date in eukaryotic cells, as it involves circa 450 protein instances of 30 different protein types, some of them present in 16 copies. The fourth one is of geometric flavor, and is concerned with the 3D embedding of the graph(s) generated. Since the nodes represent proteins and since two proteins must form a bio-physically valid interface if they touch at all, information on the shape of the proteins could be used to find plausible embeddings that would constrain the combinatorially valid solutions. This would be especially helpful to recover the edges which are known from experiments, but do not appear in exact or approximate solutions of the minimal connectivity problem.

[3] The octamer core of the human eIF3 complex comprises of the following subunits - a, c, e, f, h, k, l and m, where single letter alphabets are used to name all the subunits.

5.7 Supplemental: Statistics per Assembly

5.7.1 Yeast Exosome

Protein	Ref. Degree	Degree:#solutions							
Csl4	6	1:1644							
Dis3	4	1:1332,	2:312						
Mtr3	6	1:1130,	2:466,	3:48					
Rrp4	5	1:1188,	2:456						
Rrp40	4	1:768,	2:876						
Rrp41	4	1:290,	2:638,	3:501,	4:182,	5:31,	6:2		
Rrp42	5		2:520,	3:724,	4:334,	5:62,	6:4		
Rrp43	6		2:1370,	3:274					
Rrp45	7		2:135,	3:452,	4:575,	5:353,	6:111,	7:17,	8:1
Rrp46	5		2:1370,	3:274					

Table 5.5: **Yeast Exosome, solution set $\mathcal{S}_{\mathbf{MILP}}$: degree distribution across all the solutions, on a per protein basis.** The reference degree refers to the number of neighbors of a given protein in the assembly, based on the reference contacts. Note that a degree larger than the reference degree corresponds to false positive contacts delivered by the algorithm.

Protein	Ref. Degree	Δ_s^{left}	Δ_s^{center}	Δ_s^{right}
Csl4	6	(0, 0, 1)	(0, 1, 1)	(5, 5, 6)
Dis3	4	(0, 0, 1)	(1, 1, 2)	(2, 3, 3)
Mtr3	6	(0, 0, 0)	(1, 1, 3)	(3, 5, 5)
Rrp4	5	(0, 0, 1)	(1, 1, 2)	(3, 4, 4)
Rrp40	4	(0, 0, 0)	(1, 2, 2)	(2, 2, 3)
Rrp41	4	(0, 0, 2)	(1, 2, 4)	(0, 2, 3)
Rrp42	5	(0, 1, 2)	(1, 2, 5)	(0, 3, 4)
Rrp43	6	(0, 0, 1)	(1, 2, 3)	(3, 4, 5)
Rrp45	7	(0, 1, 2)	(2, 3, 6)	(1, 4, 5)
Rrp46	5	(0, 0, 0)	(2, 2, 3)	(2, 3, 3)

Table 5.6: **Yeast Exosome, solution set $\mathcal{S}_{\mathbf{MILP}}$: neighborhoods of a protein in all solutions versus neighborhood in the reference assembly.** The triples are defined by Eqs. (5.17, 5.18, 5.19) with respect to $\mathcal{S}_{\text{MILP}}$.

Protein	Ref. Degree	$N(p, \mathcal{S}_{\text{MILP}})\Delta_s N(p, \text{E}_{\text{Ref}})$
Csl4	6	(2, 4, 2)
Dis3	4	(1, 4, 0)
Mtr3	6	(0, 3, 3)
Rrp4	5	(2, 3, 2)
Rrp40	4	(0, 3, 1)
Rrp41	4	(2, 4, 0)
Rrp42	5	(2, 5, 0)
Rrp43	6	(3, 6, 0)
Rrp45	7	(2, 6, 1)
Rrp46	5	(0, 4, 1)

Table 5.7: **Yeast Exosome, solution set $\mathcal{S}_{\text{MILP}}$: union of neighborhoods of a protein in all solutions versus neighborhood in the reference assembly**. The triples are defined by Eq. (5.20) with respect to $\mathcal{S}_{\text{MILP}}$.

5.7.2 Yeast Proteasome Lid

Protein	Ref. Degree	Degree:#solutions				
Rpn3	6		2:95,	3:144,	4:70,	5:14, 6:1
Rpn5	5		2:65,	3:143,	4:91,	5:23, 6:2
Rpn6	4	1:252,	2:72			
Rpn7	5		2:175,	3:120,	4:27,	5:2
Rpn8	6		2:60,	3:222,	4:42	
Rpn9	3		2:240,	3:84		
Rpn11	6	1:165,	2:128,	3:29,	4:2	
Rpn12	1	1:324				
Sem1	2		2:270,	3:54		

Table 5.8: **Yeast Proteasome Lid, solution set $\mathcal{S}_{\text{MILP}}$: degree distribution across all the solutions, on a per protein basis.** The reference degree refers to the number of neighbors of a given protein in the assembly, as determined experimentally by MS, MS/MS (C_{Dim}) and through cross linking experiments (C_{XL}) and encoded in the set of contacts C_{Cryo} inferred on reconstruction of the cryo-EM map. Note that a degree larger than the reference degree corresponds to false positive contacts delivered by the algorithm.

Protein	Ref. Degree	Δ_s^{left}	Δ_s^{center}	Δ_s^{right}
Rpn3	6	(0, 0, 1)	(2, 3, 5)	(1, 3, 4)
Rpn5	5	(0, 0, 2)	(2, 3, 4)	(1, 2, 3)
Rpn6	4	(0, 0, 1)	(1, 1, 1)	(3, 3, 3)
Rpn7	5	(0, 1, 3)	(1, 2, 3)	(2, 3, 4)
Rpn8	6	(0, 0, 1)	(2, 3, 3)	(3, 3, 4)
Rpn9	3	(0, 1, 2)	(1, 1, 3)	(0, 2, 2)
Rpn11	6	(0, 0, 1)	(1, 1, 3)	(3, 5, 5)
Rpn12	1	(0, 1, 1)	(0, 0, 1)	(0, 1, 1)
Sem1	2	(0, 0, 1)	(2, 2, 2)	(0, 0, 0)

Table 5.9: **Yeast Proteasome Lid, solution set $\mathcal{S}_{\text{MILP}}$: neighborhoods of a protein in all solutions versus neighborhood in the reference assembly.** The triples are defined by Eqs. (5.17, 5.18, 5.19) with respect to $\mathcal{S}_{\text{MILP}}$.

Protein	Ref. Degree	$N(p, \mathcal{S}_{\text{MILP}})\Delta_s N(p, R)$
Rpn3	6	(1, 5, 1)
Rpn5	5	(2, 4, 1)
Rpn6	4	(1, 1, 3)
Rpn7	5	(3, 3, 2)
Rpn8	6	(1, 3, 3)
Rpn9	3	(3, 3, 0)
Rpn11	6	(1, 4, 2)
Rpn12	1	(5, 1, 0)
Sem1	2	(1, 2, 0)

Table 5.10: **Yeast Proteasome Lid, solution set $\mathcal{S}_{\text{MILP}}$: union of neighborhoods of a protein in all solutions versus neighborhood in the reference assembly**. The triples are defined by Eq. (5.20) with respect to $\mathcal{S}_{\text{MILP}}$.

5.7.3 eIF3

Protein	Ref. Degree	Degree:#solutions				
eIF3a	4	1:144,	2:36			
eIF3b	4	1:72,	2:84,	3:24		
eIF3c	4		2:120,	3:60		
eIF3d	2	1:120,	2:60			
eIF3e	3			3:48,		5:36
eIF3f	2		2:180			
eIF3g	2		2:48,	3:84,	4:42,	5:6
eIF3h	4			3:180		
eIF3i	2	1:48,	2:84,	3:42,	4:6	
eIF3k	1	1:180				
eIF3l	3		2:108,	3:72		
eIF3m	3		2:180			

Table 5.11: **eIF3, solution set $\mathcal{S}_{\text{MILP}}$: degree distribution across all the solutions, on a per protein basis.** The reference degree refers to the number of neighbors of a given protein in the assembly, as determined experimentally by MS and MS/MS and encoded in the set of contacts C_{Cryo} inferred on reconstruction of the cryo-EM map. Note that a degree larger than the reference degree corresponds to false positive contacts delivered by the algorithm.

Protein	Ref. Degree	Δ_s^{left}	Δ_s^{center}	Δ_s^{right}
eIF3a	4	(0, 1, 2)	(0, 0, 2)	(2, 4, 4)
eIF3b	4	(0, 0, 1)	(1, 1, 3)	(1, 3, 3)
eIF3c	4	(0, 1, 2)	(0, 1, 3)	(1, 3, 4)
eIF3d	2	(0, 0, 1)	(1, 1, 1)	(1, 1, 1)
eIF3e	3	(1, 1, 2)	(2, 3, 3)	(0, 0, 1)
eIF3f	2	(0, 0, 0)	(2, 2, 2)	(0, 0, 0)
eIF3g	2	(0, 1, 3)	(2, 2, 2)	(0, 0, 0)
eIF3h	4	(0, 1, 1)	(2, 2, 3)	(1, 2, 2)
eIF3i	2	(0, 1, 3)	(1, 1, 1)	(1, 1, 1)
eIF3k	1	(0, 0, 0)	(1, 1, 1)	(0, 0, 0)
eIF3l	3	(0, 0, 1)	(2, 2, 2)	(1, 1, 1)
eIF3m	3	(0, 0, 0)	(2, 2, 2)	(1, 1, 1)

Table 5.12: **eIF3, solution set $\mathcal{S}_{\text{MILP}}$: neighborhoods of a protein in all solutions versus neighborhood in the reference assembly.** The triples are defined by Eqs. (5.17, 5.18, 5.19) with respect to $\mathcal{S}_{\text{MILP}}$.

Protein	Ref. Degree	$N(p, \mathcal{S}_{\text{MILP}})\Delta_s N(p, R)$
eIF3a	4	(3, 2, 2)
eIF3b	4	(1, 3, 1)
eIF3c	4	(3, 4, 0)
eIF3d	2	(1, 1, 1)
eIF3e	3	(5, 3, 0)
eIF3f	2	(0, 2, 0)
eIF3g	2	(3, 2, 0)
eIF3h	4	(2, 3, 1)
eIF3i	2	(3, 1, 1)
eIF3k	1	(0, 1, 0)
eIF3l	3	(1, 2, 1)
eIF3m	3	(0, 2, 1)

Table 5.13: **eIF3, solution set $\mathcal{S}_{\text{MILP}}$: union of neighborhoods of a protein in all solutions versus neighborhood in the reference assembly**. The triples are defined by Eq. (5.20) with respect to $\mathcal{S}_{\text{MILP}}$.

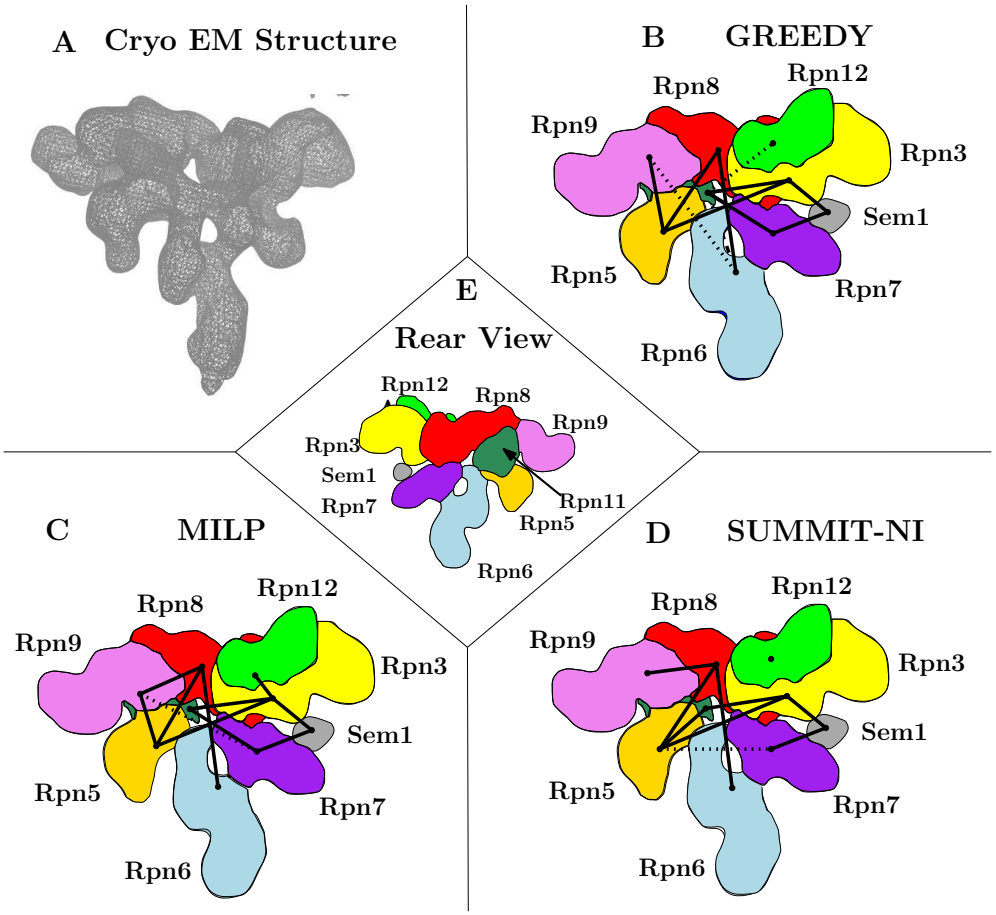
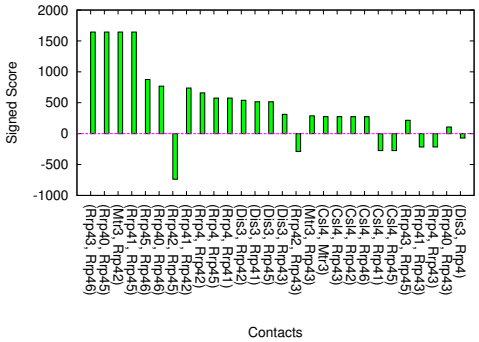
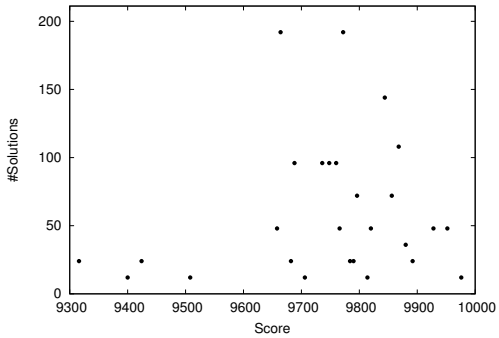


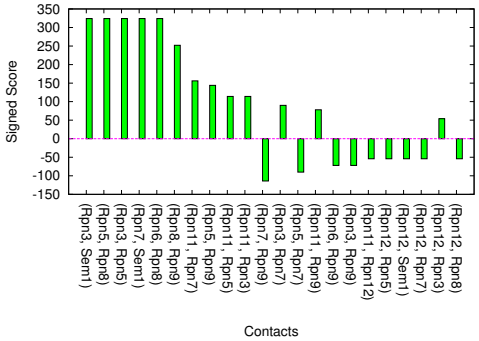
Figure 5.5: **Yeast 19S Proteasome Lid: contacts computed by the algorithms.** **(A)** Isosurface from the cryo-EM density reconstruction [LEM+12, Fig. 2]. **(B,C,D)** Sketch of the view same as that of the isosurface following delineation for each subunit and decorated with one edge per contact found by the respective resolution methods. The contacts for `MILP` correspond one of the consensus solutions. The conventions for contact classification match those of Fig. 5.4. Note, that the subunit Rpn11 is hidden in between the subunits Rpn5, Rpn8, Rpn9. It is also to be noted that the cryo-EM based reconstruction of [LEM+12] features 8 out of 9 subunits — the small Sem1 subunit was not detected in the SDS-PAGE. It is therefore placed based on cross-linking experiments [STA+06]. **(E)** Sketch of the rear view w.r.t. to those in **B,C,D**.



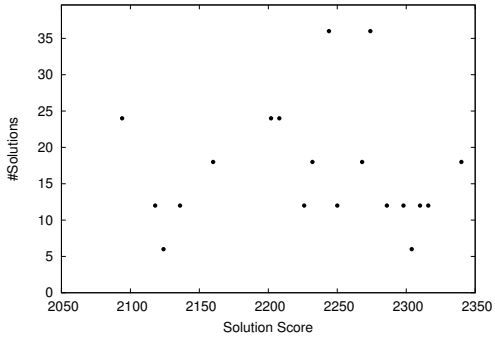
(a) Yeast Exosome: signed contact scores.



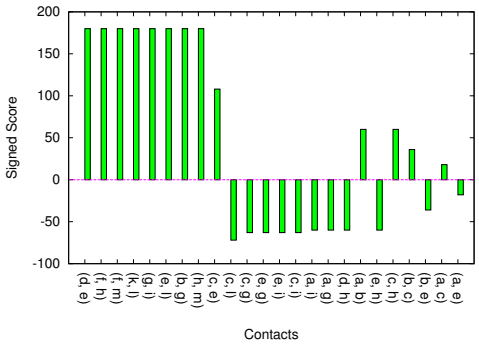
(b) Yeast Exosome: distribution of scores.



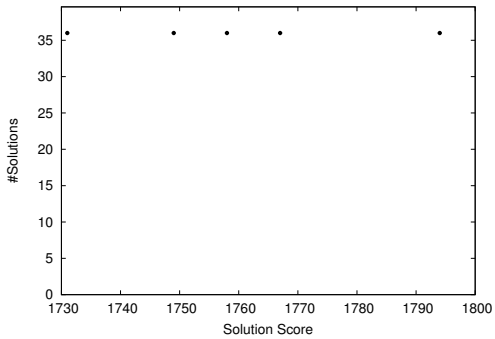
(c) Yeast Proteasome Lid: signed contact scores.



(d) Yeast Protasome Lid: distribution of scores.

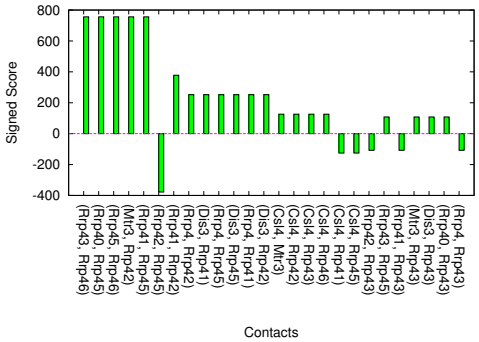


(e) eIF3: signed contact scores.

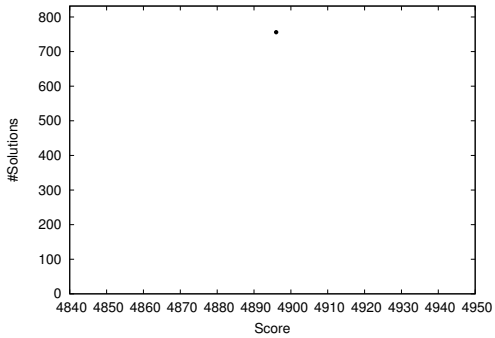


(f) eIF3: distribution of scores.

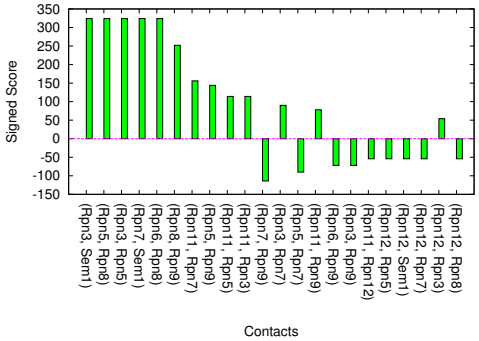
Figure 5.6: **Algorithm** `MILP`: analysis of solutions in $\mathcal{S}_{\text{MILP}}$. Note that plots on the left column show all the contacts in the union of solutions.



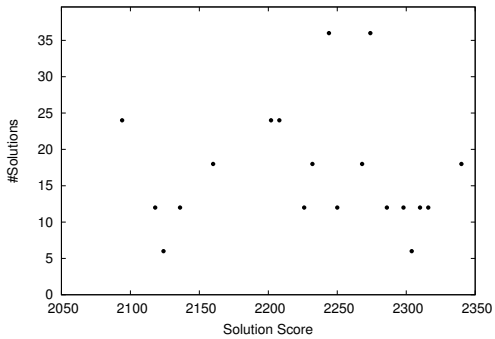
(a) Yeast Exosome: signed contact scores.



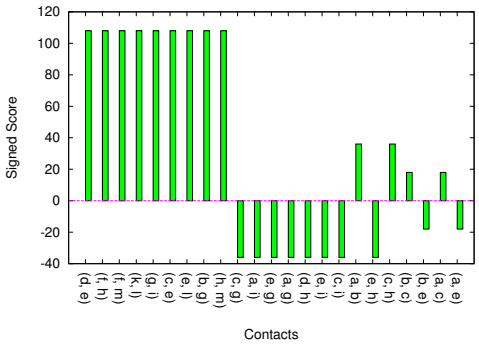
(b) Yeast Exosome: distribution of scores.



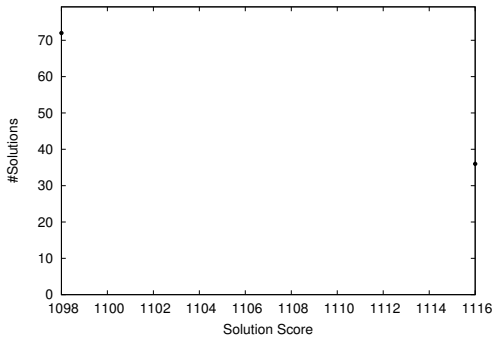
(c) Yeast Proteasome Lid: signed contact scores.



(d) Yeast Protasome Lid: distribution of scores.



(e) eIF3: signed contact scores.



(f) eIF3: distribution of scores.

Figure 5.7: **Algorithm** `Greedy`: analysis of solutions in $\mathcal{S}_{\text{Greedy}}$. Note that plots on the left column show all the contacts in the union of solutions.

Chapter 6

Connectivity Inference: the Weighted Case

6.1 Connectivity Inference from Sets of Oligomers

Structural inference from oligomers and contacts. Unraveling the function of macro-molecules and macro-molecular machines requires atomic level data, both in their static and dynamic dimensions, the latter coding for thermodynamic and kinetic properties [SXK+13]. However, obtaining even static snapshots of large systems remains a *tour de force*, so that alternative methods are being developed, based in particular on *reconstruction by data integration* (RDI), a strategy aiming at producing models of assemblies using complementary experimental data [ADV+07]. In its full generality, RDI accommodates both structural and purely combinatorial data [AFK+08]. The former typically consists of crystallographic (high resolution) structures, electron microscopy maps, and NMR models. The latter comprise information on the composition and copy numbers of subunits, as well as pairwise contacts. Given a large assembly, information on oligomers (i.e., sub-complexes of the assembly) can be obtained by methods such as tandem affinity purification [ea01] or native mass spectrometry [SR07, RDD+12], and such oligomers can be complemented by information on pairwise protein - protein interactions [BLJ07, BTD+08]. More specifically, oligomers of varying size can be obtained under various experimental conditions. While stringent conditions (e.g. low pH) result in complete dissociation of the assembly, so that the individual molecules are identified, less stringent conditions result in the disruption of the assembly into multiple overlapping oligomers. Assembled together, such oligomers can be used to infer contacts within the assembly [SR07]. In the context of RDI, the models stemming from such analysis do not, in general, achieve atomic resolution. They can, however, be used to bridge the gap to atomic level models of sub-systems of the assembly under scrutiny [DDC12, DDC13].

Unweighted and weighted connectivity inference: MCI and MWCI. Consider a macro-molecular assembly consisting of subunits (typically proteins or nucleic acids). Assume that these subunits are known, but that the pairwise contacts between them are unknown or only partially known – in this latter case the presence or the likelihood of selected contacts is known. Connectivity Inference (CI) is the problem concerned with the elucidation of contacts between these subunits, as it ideally aims at producing one contact for each pair of subunits sharing an interface in the assembly (Fig. 6.1). Note that mathematically, the subunits may be seen as the nodes of a graph whose edges are defined by the contacts. Thus, in the sequel, we use contacts and edges interchangeably.

To address CI, let an *oligomer formula* be a list of subunits defining a connected component within the assembly. That is, an oligomer formula is the description of the composition of the oligomer, giving the number of copies of each molecule. We define a *connectivity inference specification* (specification for short) as a list of oligomers. The solution of a CI problem consists of a set of contacts, denoted S in the sequel. This set is called *a valid edge set* or a *solution* provided that for each oligomer and also for the whole complex:

restricting the edges from S to the vertices of an oligomer formula yields a connected graph (Fig. 6.1). In defining valid solutions, two critical questions arise: how many contacts should one seek, and should all edges be treated on an equal footing.

On the number of contacts. In the absence of a priori knowledge on the likelihood of individual contacts, a solution is naturally assessed by its number of contacts. Mastering this size is non trivial, since the number of interfaces between subunits in the assembly is unknown. On the one hand, the trivial solution involving all possible edges is uninteresting since it is likely to contain a large number of false positives. On the other hand, one may solve the *Minimum Connectivity Inference* problem (MCI), namely the variant of CI minimizing the number of contacts used. To do so, observe that minimally connecting an oligomer merely requires a tree, that is a graph whose number of edges is the number of vertices (subunits) minus one. Thus, solving MCI consists of choosing for each oligomer the tree yielding the solution of minimum size. Yet, in doing so, one is likely to generate false negatives. Given these two extremes, one goal of this work is to optimize the number of contacts reported, so as to maximize the number of true positives and true negatives – a goal that will be achieved using so-called consensus edges and a bootstrapping strategy.

A priori knowledge on contacts. known. On the experimental side, various assays have been developed to check whether two proteins interact, including yeast-two-hybrid, mammalian protein-protein interaction trap, luminescence-based mammalian interactome, yellow fluorescent protein complementation assay, co-immuno precipitation, etc [BLJ07, BTD$^+$08]. But information obtained must be used with care for several reasons, notably because expression systems force promiscuity between proteins which may otherwise be located in different cellular compartments, and also because affinity purification typically involves concentration beyond physiological levels. On the *in-silico* side, various interactions attributes can be used, such as gene expressions patterns (proteins with identical patterns are more likely to interact), domain interaction data (a known interaction between two domains hints at an interaction between proteins containing these domains), common neighbors in protein - protein interaction networks, or bibliographical data (number of publications providing evidence for a particular interaction). Here again, these pieces of information have a number of caveats. In particular, structural data from crystallography or mass spectrometry yield a bias towards stable interactions, at the detriment of transient ones. For these reasons, strategies computing confidence scores usually resort to machine learning tools trained on the aforementioned data [YMJ12] and also [TRT$^+$10].

In any case, being able to accommodate such a previous knowledge is precisely another goal of the algorithms developed in this work.

Algorithms. From a computer science perspective, solving CI problems is a hard task (supplemental section 6.9). Two algorithms targeting such problems have been developed so far.

The first one is a two-stage heuristic method [THS$^+$08]. First, random graphs meeting the connectivity constraint are generated, by incrementally adding random edges. Second, a genetic algorithm is used to reduce the number of edges, and also boost their diversity. Once the average size of the graphs stabilizes, the pool of graphs is analyzed to spot highly conserved edges.

The second one is our method solving MCI problems, based on a mixed integer linear program [AAC$^+$13]. On the one hand, this work delineates the combinatorial hardness of the CI problem, and offers two algorithms. Of the particular interest is `MILP`, since it delivers all optimal solutions of a given MCI problem. (Following our discussion above, note that this algorithm may require exponential time for hard instances, event though this behavior was not observed for the cases processed.) On the other hand, when assessed against contacts seen in crystal structures, the solutions of `MILP` suffer from two limitations. First, in all solutions, few false negatives are observed, at the expenses of selected false positives. On the other hand, since all edges have a unit weight, one cannot favor or penalize some of them.

In this context, this paper makes two improvements. First, we introduce the *Minimum Weight Connectivity Inference* problem (MWCI), which allows computing optimal solutions incorporating a priori knowledge

on the likelihood of edges. Second, we present algorithm `MILP-W` to solve MWCI problems, an algorithm aiming at maximizing the sensitivity and specificity of the set of contacts reported.

6.2 Minimum Weight Connectivity Inference: Mathematical Model

Oligomers and pools of edges. In solving CI problems, a valid edge set consists of edges such that each of them involves two subunits belonging to at least one oligomer. More precisely, consider an oligomer O_i. This oligomer defines a pool of candidate edges equal to all pairs of subunits found in O_i. Likewise, the pool of candidate edges $\mathrm{Pool_E}(\mathcal{O})$ defined by a set of oligomers $\mathcal{O}$ is obtained by taking the union of the pools defined by the individual oligomers. Note that one can also consider a restricted set of oligomers involving the oligomers whose size is bounded by an integer s, denoted $\mathcal{O}_{\leq s}$, the corresponding pool of candidate edges being denoted $\mathrm{Pool_E}(\mathcal{O}_{\leq s})$. The rationale for using small oligomers is that they favor local contacts. (Note that the extreme case is that of a dimer, since the contact seen in a dimer must belong to every solution.) Note also that one can edit a pool of edges, to enforce or forbid a given edge in all solutions. For example, if a cryo-electron microscopy map of the assembly is known and two proteins have been located far apart in the map, one can forbid the corresponding contact even though the two proteins appear in a common oligomer.

We now present two ways to solve CI problems.

Unweighted case. In the unweighted case, each edge from the pool is assigned a unit weight, so that the weight of a solution is the number of its edges. The corresponding optimization problem is called MCI, and an algorithm solving it, `MILP`, has been proposed in [AAC+13].

Weighted case. In the weighted case, each candidate edge e from the pool $\mathrm{Pool_E}(\mathcal{O})$ is assigned a weight $w(e)$, namely a real number in range $[0, 1]$. This number encodes the likelihood for the edge to be a true contact. Taking $G = 1/2$ as a baseline (i.e. no a priori on this contact), a value $F > G$ is meant to favor the inclusion of this edge in solutions, while a value $U < G$ is meant to penalize this edge.

Unifying the unweighted and weighted cases: MWCI problems. Depending on how much information is available on candidate contacts, one may wish to stress the number of contacts in a solution, or their total weight. Both options can actually be handled at once by *interpolating* between the previous two problems. Using a real number $\alpha \in [0, 1]$, we define a functional mixing the number of edges and their weights, this latter one being favored for values beyond the threshold 1/2. That is, we define the *cost* of a solution S using two terms respectively corresponding to the number of edges and their weights:

$$C(S) = \alpha \sum_{e \in S} 1 + (1 - \alpha) \sum_{e \in S} (1/2 - w(e)) = \sum_{e \in S} C_\alpha(e), \tag{6.1}$$

with

$$C_\alpha(e) = \frac{\alpha + 1}{2} - (1 - \alpha) w(e). \tag{6.2}$$

Eq. (6.1) corresponds to the objective of the optimization problem denoted MWCI in the sequel.

The following comments are in order:

- In using $\alpha = 1$, which is the strategy used by algorithm `MILP` [AAC+13], the weights play no role, and the inter-changeability of edges favors the exploration of a large pool of solutions.
- The situation is reversed for small values of α. In particular, the conjunction $\alpha < 1$ and different weights for all edges typically yields a small number of solutions, since ties between solutions are broken by the weights.

- A null weight does not prevent a given edge to appear in solutions. To forbid an edge, one should edit the pool of candidate edges, as explained above i.e. remove this edge from the pool.

Remark 6. *Assume that each edge has a default weight d instead of 1/2. Eq.* (6.1) *is a particular case of the following*

$$C_{\alpha,d}(e) = \alpha(1-d) + d - (1-\alpha)w(e). \tag{6.3}$$

Setting $d = 1/2$ in Eq. (6.3) *yields the edge cost of Eq.* (6.1)*. On the other hand, setting $d = 1$ yields a constant term 1 instead of $(\alpha+1)/2$. Since the default $d = 1$ yields a weighting criterion less sensitive to weights, we use $d = 1/2$.*

We also observe that $dC_{\alpha,d}(e)/d\alpha = 1 - d + w(e)$. Thus, when varying α, the edge weight prevails or not depending on its value with respect to the value $1-d$. For $d = 1/2$, one gets $dC_{\alpha,d}(e)/d\alpha = 1/2 + w(e)$.

6.3 Minimum Weight Connectivity Inference: Algorithms

6.3.1 Algorithm `MILP-W`

Algorithm `MILP-W` generalizes the unweighted version `MILP` [AAC+13], and allows enumerating all optimal solutions with respect to the criterion of Eq. (6.1). The algorithm solves a mixed integer linear program, using constraints imposing the connectivity constraints inherent to all oligomers. Candidate edges are represented by binary variables taking the value 1 when edges belong to a specific solution [AAC+13] and 0 otherwise.

More precisely, algorithm `MILP-W` iteratively generates all optimal solutions, and adds at each iteration extra constraints preventing from finding the same solution twice. To this end, the method starts with a first resolution of the problem to get an optimal solution, if any. This solution defines a set of edges and the associated value OPT for the criterion of Eq. (6.1). To check whether another solution matching OPT exists, a new constraint preventing the concomitant selection of all edges from the first solution is added. More formally, the sum of the binary variables associated with the solution just produced is forced to be strictly less than the number of edges in solutions seen so far. The resolution is launched again, and the criterion value is compared to OPT. This process is iterated until the value of the solution exceeds OPT.

Remark 7. *By picking the adequate combination of α and $w(\cdot)$, the individual edge cost of Eq.* (6.2) *can be null. Edges with null cost can create troubles in the enumeration problem, since solutions with the same cost but nested sets of edges can be created. To get rid of spurious large edges, it is sufficient to build the Hasse diagram (for the inclusion) of all solutions, and remove the terminal nodes of this diagram.*

6.3.2 Solutions and consensus solutions

The set of all optimal solutions reported by `MILP-W` is denoted $\mathcal{S}_{\text{MILP-W}}$, and the set of contacts used in these solutions is denoted $\mathcal{E}_{\text{MILP-W}}$. The *size of a solution* $S \in \mathcal{S}_{\text{MILP-W}}$, denoted $|S|$, is its number of contacts. The *score of a contact* appearing in a solution $S \in \mathcal{S}_{\text{MILP-W}}$, called *contact score* for short, is the number of solutions from $\mathcal{S}_{\text{MILP-W}}$ containing it. The *score of a solution* $S \in \mathcal{S}_{\text{MILP-W}}$ is the sum of the scores of its contacts. A *consensus solution* is a solution achieving the maximum score over $\mathcal{S}_{\text{MILP-W}}$. The set of all such solutions being denoted $\mathcal{S}^{cons.}_{\text{MILP-W}}$. The contacts found in consensus solutions are called the *consensus contacts*, and define the set $\mathcal{E}^{cons.}_{\text{MILP-W}}$.

As noticed earlier, when $\alpha = 1$, algorithm `MILP-W` matches algorithm `MILP`. Therefore, for the sake of clarity, the solution set, consensus solutions and the associated edge sets are respectively denoted $\mathcal{S}_{\text{MILP}}, \mathcal{S}^{cons.}_{\text{MILP}}, \mathcal{E}_{\text{MILP}}$ and $\mathcal{E}^{cons.}_{\text{MILP}}$. These notations are summarized in Table 6.1.

To further assess the quality of the solution set $\mathcal{S}(= \mathcal{S}_{\text{MILP}}, \mathcal{S}_{\text{MILP-W}})$, assume that a reference set of contacts E_{Ref} is known. The ideal situation is that where a high resolution crystal structure is known, since then, all pairwise contacts can be inferred [LC10]. This reference set together with the pool $\text{Pool}_{\text{E}}(\mathcal{O})$ define positive (P), negative (N), and missed contacts (M) (Fig. 6.2). From these groups, one further classifies the

edges of a predicted solution in set $\mathcal{S}$ into four categories, namely true positive (TP), false positive (FP), true negative (TN), and false negative (FN).

Positives (P) and negatives (N) decompose as $P = TP + FN$, and $N = TN + FP$, from which one defines the sensitivity $\mathrm{ROC}_{sens.}$ and the specificity $\mathrm{ROC}_{spec.}$ as follows:

$$\mathrm{ROC}_{sens.} = \frac{|TP|}{|P|}, \qquad \mathrm{ROC}_{spec.} = \frac{|TN|}{|N|}. \tag{6.4}$$

Note that specificity requires the set N to be non empty, which may not be the case if $\mathrm{Pool}_{\mathrm{E}}(\mathcal{O}) \subset \mathrm{E}_{\mathrm{Ref}}$.

We also combine the previous values to define the following *coverage score*, which favors true positives, penalizes false positives and false negatives, and scales the results with respect to the total number of reference contacts (since P might be included into $\mathrm{E}_{\mathrm{Ref}}$ if the pool size is too small):

$$\mathrm{Cvg}(\mathcal{S}) = \frac{|TP| - (|FP| + |FN|)}{|\mathrm{E}_{\mathrm{Ref}}|} \tag{6.5}$$

Note that the maximum value is one, and that the coverage score may be negative.

6.3.3 Algorithm MILP-$\mathrm{W_B}$

The focus on consensus edges is quite natural, since these may prosaically be seen as the *backbone* of the connectivity in the assembly. However, alternative edges of significant importance may exist too. To unveil such edges, we preclude one or more consensus edges, so as to trigger a rewiring of the connectivity of solutions, and check which novel consensus edges appear along the way. Implementing this strategy requires two precautions, namely: (i) edges corresponding to dimers must be kept for a solution to be valid, and (ii) hindering too many edges may yield a connectivity inference problem without any solution.

More precisely, we start precluding the consensus contacts i.e. the initial consensus contacts $\mathcal{E}^{cons.}_{\text{MILP-W}}$ minus the dimers, one at a time from the pool of contacts to be explored. We subsequently report the union of consensus contacts (including the initial consensus contacts which we began with) yielded after all the MILP-W runs. The process can be iterated by precluding two or more contacts at a time. This strategy triggers rewiring of the system to a greater extent, at the risk of inducing more false positives. (See pseudo code in the supplemental section 6.9.)

6.4 Material: Test Systems

We test the performance of the algorithms MILP-W and MILP-$\mathrm{W_B}$ on the following three systems for which reference contacts for validation are available either coming from crystal structure or from various biophysical experiments such as cryo-EM based reconstruction, cross-linking, and MS/MS dimers. See supplemental section 8.1 for the input to the algorithms and the supplemental section 8.2 for the reference contacts.

6.4.1 Yeast exosome

The exosome involves 10 protein types (Figs. 6.3 and 6.4), and 19 oligomers [1] have been reported [THS+08], ranging in size from two to nine (Table 6.2 and supplemental section 8.1).

Oligomers up to size five are required to encompass 9 out of 10 proteins — the protein Csl4 is present in size nine oligomers only. In terms of contacts, classical interfaces modeling tools [LC10] applied to the crystal structure yield 26 contacts amidst the 10 proteins, and 20 contacts in the assembly depleted of Csl4 (Fig. 6.4).

The status of Csl4 is interesting, since, as discussed in section 6.2, local contacts are favored by small oligomers. In the sequel, we therefore consider two settings, namely the full exosome, and the exosome

[1] Originally 21 oligomers are reported including a trivial case of having all 10 protein types and one other oligomer of size 8 has a duplicate, leaving behind 19 distinct oligomers.

without Csl4. In the former case, all oligomers define a pool $\mathrm{Pool_E}(9)$ of 45 candidate edges; in the latter, the pool $\mathrm{Pool_E}(8)$ contains 36 candidate edges.

6.4.2 Yeast 19S Proteasome lid

Proteasomes are protein assemblies involved in the elimination of damaged or misfolded proteins, and the degradation of short-lived regulatory proteins. The most common form of proteasome is the 26S, which involves two filtering caps (the 19S), each cap involving a peripheral lid, composed of 9 distinct protein types each with unit stoichiometry (Fig. 6.9).

Series of overlapping oligomers were formed by mass spectrometry (MS), tandem MS and cross-linking using BS3. In total, 14 complexes were obtained out of which 8 came from MS, MS/MS and 6 came from cross-linking experiments [STA+06] (Table 6.3 and supplemental section 8.1).

6.4.3 Human eIF3

Eukaryotic initiation factors (eIF) are proteins involved in the initiation phase of the eukaryotic translation. They form a complex with the 40S ribosomal subunit, initiating the ribosomal scanning of mRNA. Among them, human eIF3 consists of 13 different protein types each with unit stoichiometry (Fig.6.12). The eIF3 complex in this text refers to the human eIF3 unless otherwise stated.

A total of 27 complexes were generated from the assembly by manipulating the ionic strength of the solution and using tandem mass spectrometry [ZSF+08] (Table 6.4 and supplemental section 8.1). The subunit eIF3j is labile and since none of 27 subcomplexes comprises of this subunit we exclude it from the list of protein types, leaving behind 12 protein types.

6.5 Results

We first provide results for MILP-W_B with $\alpha = 1$, namely when all edges have the same weight. In a second step, we illustrate the benefits of using weights.

6.5.1 Algorithm MILP-W_B

As explained in section 6.3.3, algorithm MILP-W_B works by accumulating consensus contacts (contacts from highest scoring solutions) from MILP-W and those due to local rewiring as a result of precluding the initial consensus contacts one (or more) at a time. Consequently, our analysis focuses on the sensitivity, specificity and coverage statistics introduced in section 6.3.2 in four settings:

- (C-1) The statistics for the edge set $\mathcal{E}_{\text{MILP}}$, which serve as a baseline.
- (C-2) The statistics for the edge set $\mathcal{E}^{cons.}_{\text{MILP}}$.
- (C-3) The statistics for the edge set $\mathcal{E}_{\text{MILP-W}_\text{B}}$ returned by MILP-W_B after one iteration.
- (C-4) the statistics for the edge set $\mathcal{E}_{\text{MILP-W}_\text{B}}$ obtained when algorithm MILP-W_B terminates.

The results are presented on Figs. 6.5, 6.6, 6.7 and 6.8 for the yeast exosome, Figs. 6.10 and 6.11 for the yeast proteasome, and Figs. 6.13 and 6.14 for human eIF3.

The following consistent observations can be made for the three systems:

- In comparing (C-1) against (C-2), the sensitivity decreases since consensus solutions have fewer edges. On the other hand, the specificity increases, indicating a large number of true negatives or equivalently a small number of false positives – an observation in line with the high scores of edges in consensus solutions.
- In comparing (C-3) against (C-4), the sensitivity increases, while the specificity decreases. The variations observed for sensitivity and specificity actually depends on the number of contacts precluded in the bootstrap procedure. Indeed, precluding more contacts triggers more rewiring, which in turns yields a larger set of true positive edges (increased sensitivity), at the expense of more false positives (decreased specificity).
- Finally, for algorithm MILP-W_B, one observes that a small number of iterations, typically in the range $1, \ldots, 3$, is favorable to high coverages. This owes to the aforementioned counterbalance between sensitivity and specificity. In particular, when the bootstrap procedure halts, all contacts from the pool have been used, which entails a large number of true positives – high sensitivity, but also a large number of false positive – low specificity. Thus, the user may choose the risk level (in terms of false positives) he/she is willing to accepts, depending on whether the focus is on sensitivity or specificity.

Comparison to previous work. The statistics just discussed compare favorably to previous work, obtained in particular with the heuristic network inference algorithm [THS$^+$08]. We illustrate this fact with the results produced by MILP-W_B after one iteration.

On the yeast exosome with Csl4, the sensitivity of MILP-W_B is ~ 1.67 (=0.77/0.46) times that of network algorithm and *Cvg.* score increases from -0.08 to 0.35 (Figs. 6.5 and 6.6; lines T3 vs T0 in the supplemental Table 6.5). See also Figs. 6.7 and 6.8 for the exosome without Csl4.

For the yeast proteasome, one observes that the sensitivity for $\mathcal{E}_{\text{MILP-W}_\text{B}}$ is 1.76 (=0.74/0.42) times that published earlier [THS$^+$08]. Also, *Cvg.* score increases from -0.21 to 0

(Figs. 6.10 and 6.11; lines T3 vs T0 in the supplemental Table 6.8).

The comparison is not possible, for eIF3, though, since the previously published contacts were computed manually using experimental information from various other sources [ZSF$^+$08]. See however Figs. 6.13 and 6.14, and the supplemental Table 6.10) for the results using our algorithms.

6.5.2 Algorithm MILP-W

In this section, we illustrate the role of weights stemming from using Eq. (6.1) with $\alpha \neq 1$. One naturally expects a benefice in penalizing an edge which is a negative contact and which is thus predicted as a false positive or a true negative. But an improvement of statistics can also happen by merely favoring positive contacts. As an illustration, we consider the yeast exosome without Csl4 (specifications in Table 6.2), using $\alpha = 0.25$. We assign a weight of 0.6 to the following three contacts - (Rrp45, Rrp46), (Rrp40, Rrp46) and (Rrp41, Rrp42), the remaining contacts having the default weight of 0.5.

Upon moving from the instance without weights (i.e., $\alpha = 1$) to the instance with weights (i.e., $\alpha = 0.25$), we consider the changes in the sensitivity, specificity and coverage, namely:

$$\Delta = (\Delta\mathrm{ROC}_{sens.}, \Delta\mathrm{ROC}_{spec.}, \Delta\mathrm{Cvg}). \tag{6.6}$$

Consider first $\mathcal{E}_{\mathrm{MILP}}$. One observes 4 false positives instead of 5 (supplemental Fig. 6.15), improving the assessment tuple by (0, 0.06, 0.05). Thus, while none of the contacts has a weight less than 0.5, the relative value of weights has an incidence on the outcome.

Consider now $\mathcal{E}^{cons.}_{\mathrm{MILP}}$. The union of consensus contacts has no false positive, and true positives are increased from 9 to 13, improving the assessment tuple by (0.2, 0.06, 0.45) (supplemental Fig. 6.16). For $\mathcal{E}_{\mathrm{MILP\text{-}W_B}}$, the change in assessment tuple for $\mathcal{E}_{\mathrm{MILP\text{-}W_B}}$ (1st iteration) is (0.05, -0.12, 0). When more contacts are precluded, the trend seen is similar to earlier cases (supplemental Fig. 6.17).

The reader is referred to the supplemental section 6.10 for a thorough assessment obtained upon varying the weights and the value of α.

6.6 Discussion and Outlook

For these reasons, strategies computing confidence scores usually resort to machine learning tools trained on the aforementioned data [YMJ12] and also [TRT+10].

By giving access to a list of overlapping oligomers of a given macro-molecular assembly, native mass spectrometry offers the possibility to infer pairwise contacts within that assembly, opening research avenues for systems beyond reach for other structural biology techniques. In this context, our work makes three contributions, based on state-of-the art combinatorial optimization techniques.

First, we introduce the *Minimum Weight Connectivity Inference* problem (MWCI), which generalize the *Minimum Connectivity Inference* problem, by introducing weights associated with putative contacts. Second, we develop algorithm MILP-W to solve MWCI problems, taking into account a priori biological knowledge on the likelihood of contacts. Third, we also develop algorithm $\mathtt{MILP\text{-}W_B}$, a bootstrap strategy aiming at enriching the solutions reported by MILP-W. Algorithm $\mathtt{MILP\text{-}W_B}$ accumulates consensus contacts from MILP-W and those arising due to local rewiring as a result of precluding the initial consensus contacts one (or more) at a time. Our algorithms predict contacts with high specificity and sensitivity, yielding a very significant improvement over previous work, typically a twofold increase in sensitivity. Despite the combinatorial complexity of the problems addressed, all runs of algorithm MILP-W terminated within a hand-full of seconds for all the cases processed in this work. Calculations with algorithm $\mathtt{MILP\text{-}W_B}$ are more demanding, though, since the run-time depends on the combinatorics of the tuples to be precluded.

These algorithms raise a number of opportunities and challenges.

In the context of native mass spectrometry, they offer the possibility to test various parameter sets, in particular regarding the number of contacts and their likelihood, and to compare the solutions obtained. More broadly, the ability to take into account confidence levels on putative edges should be key to incorporate scores currently being designed in proteomics, in conjunction with various assays.

In terms of challenges, fully harnessing these algorithms raises difficult questions. On the practical side, one current difficulty is the lack of cases to learn from, namely assemblies for which a significant list of oligomers is known, and a high resolution structure has been obtained. Such cases would be of high interest to tune the balance between the aforementioned two criteria (number of contacts and their likelihood).

This would also aid in carrying out an in-depth study of incidence of weights on the solutions obtained from `MILP-W` runs, given true positives and false positives in the pool of contacts.

Unfortunately, mass spectrometry studies are typically attempted on assemblies whose high resolution structure is unknown and is likely to remain so, at least in the near future. On the theoretical side, outstanding questions remain open. The first one deals with the relationship between the set of oligomers processed and the solutions generated. Ideally, one would like to set up a correspondence between equivalence classes of oligomers yielding identical solutions. The ability to do so, coupled to the understanding of which oligomers are most likely generated, would be of invaluable interest. The second one relates to the generalization of our algorithms to accommodate cases where multiple copies of sub-units are present. However, the multiple copies complicate matters significantly, so that novel insights are called for not only computing solutions, but also representing them in a parsimonious fashion.

In any case, we anticipate that the implementations of our algorithms, will prove its interest for the growing community of biologists using native mass spectrometry.

6.7 Artwork

6.7.1 Methods

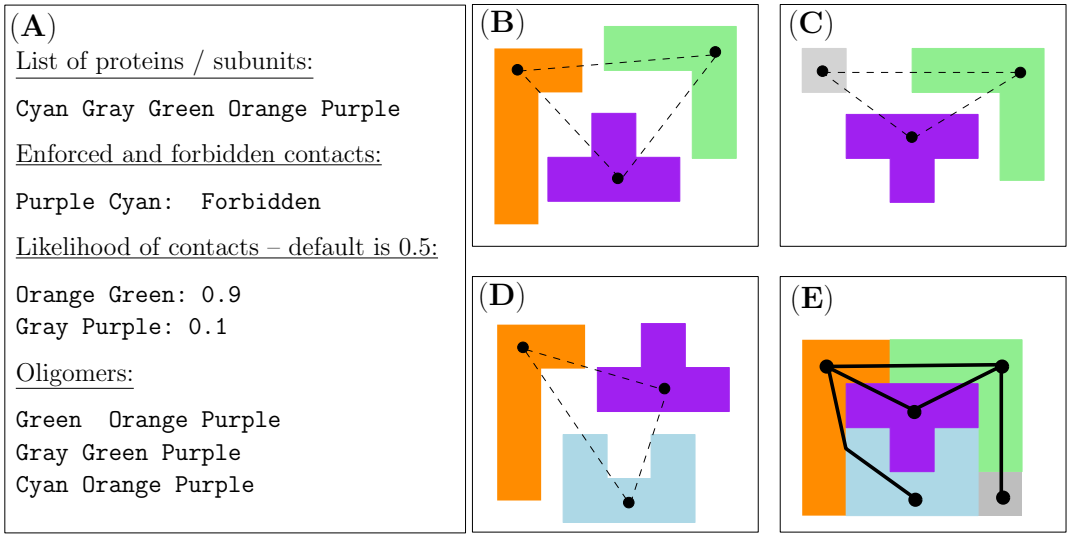


Figure 6.1: **(Minimum) Connectivity Inference from oligomers and a-priori information on contacts: illustration on a fictitious system.** Given an assembly whose subunits are known but pairwise contacts are not, and for which the composition of a number of oligomers in terms of subunits is also known, the problem consists of inferring contacts between subunits. We consider a toy example involving 5 proteins and three oligomers (three trimers), as seen on panel **(A)**). As additional information, one may enforce and/or forbid contacts, and one may also weight contacts, depending on their likelihood. To connect each oligomer using as few edges as possible, two edges must be chosen, out of three possible (panels **(B, C, D)**). The *Minimum Connectivity Inference* consists of finding the overall smallest number of edges such that each each oligomer gets connected. Panel (**(E)**) shows a solution with 6 edges (bold edges). Note that these six edges from a subset of all pairwise contacts.

	solutions	edges	consensus solutions	consensus edges
`MILP`	$\mathcal{S}_{\text{MILP}}$	$\mathcal{E}_{\text{MILP}}$	$\mathcal{S}^{cons.}_{\text{MILP}}$	$\mathcal{E}^{cons.}_{\text{MILP}}$
`MILP-W`	$\mathcal{S}_{\text{MILP-W}}$	$\mathcal{E}_{\text{MILP-W}}$	$\mathcal{S}^{cons.}_{\text{MILP-W}}$	$\mathcal{E}^{cons.}_{\text{MILP-W}}$
`MILP-W`$_{\text{B}}$	$\mathcal{S}_{\text{MILP-W}_{\text{B}}}$	$\mathcal{E}_{\text{MILP-W}_{\text{B}}}$	NA	NA

Table 6.1: **Notations for (consensus) solutions and (consensus) edges returned by the algorithms `MILP`, `MILP-W` and `MILP-W`$_{\mathbf{B}}$.**

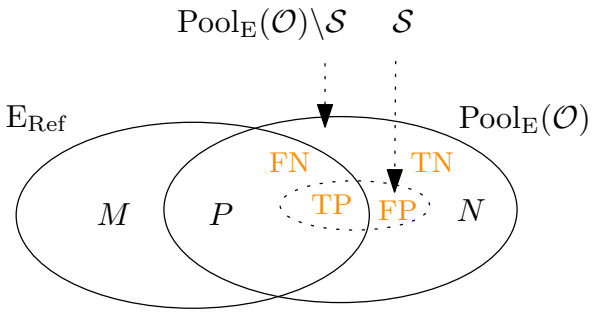


Figure 6.2: **A pool of candidate $\text{Pool}_{\text{E}}(\mathcal{O})$ and a set of reference contacts E_{Ref} define positive (P), negative (N), and missed contacts (M).** Upon performing a prediction $\mathcal{S}$, $\mathcal{S}$ and its complement $\text{Pool}_{\text{E}}(\mathcal{O})\backslash\mathcal{S}$ further split into true/false $\times$ positives/negatives (TP, FP, TN, FN).

6.7.2 Yeast Exosome

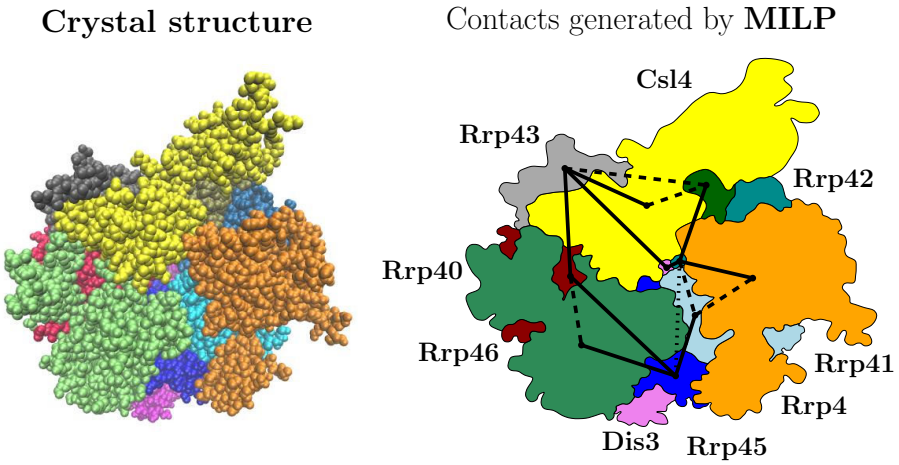


Figure 6.3: **The yeast exosome, an assembly consisting of 10 subunits.** The Connectivity Inference problem consists of inferring contacts between the subunits from the composition of oligomers, i.e. connected blocks of the assembly. **(Left)** Crystal structure **(Right)** The solid edges reported by the algorithm `MILP`, while the dashed edges are not present in the solution.

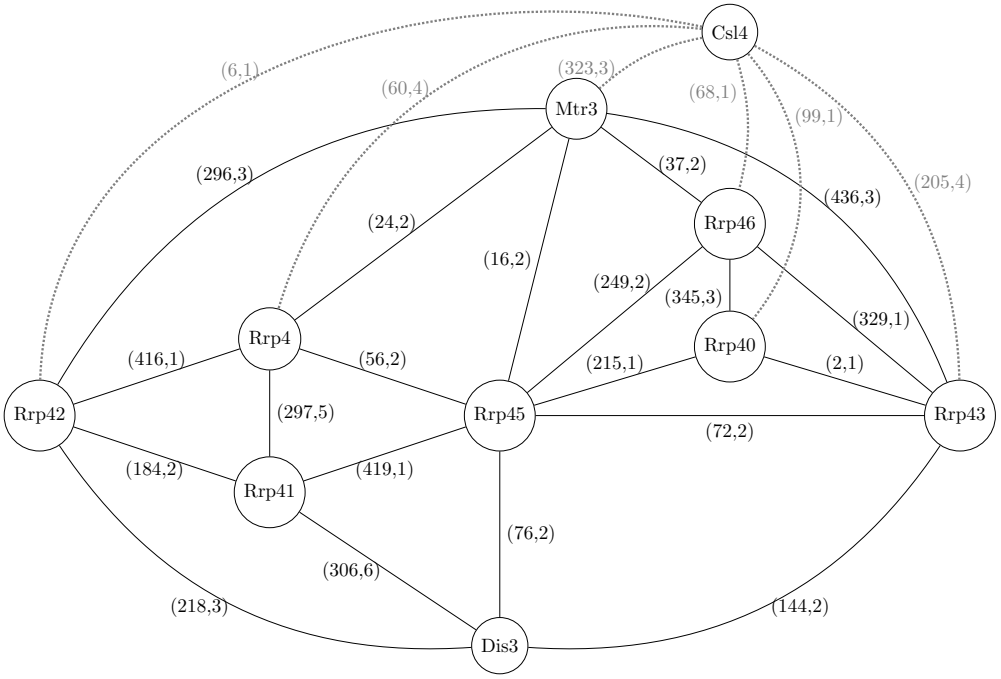


Figure 6.4: **Yeast exosome with and without Csl4.: contacts between subunits.** Each edge corresponds to an interface between two subunits. The two numbers decorating an edge respectively refer to the number of atoms involved at that interface, and to the number of patches (connected components) of the interface. Interfaces were computed with the program `intervor`, which implements the Voronoi model from [LC10]. Note that a given subunit makes from three (e.g. Rrp40) to seven (e.g. Rrp45) interfaces.

Oligomer size s	$\lvert\mathcal{O}_{\leq s}\rvert$	$\lvert\mathrm{Pool}_{\mathrm{E}}(\mathcal{O}_{\leq s})\rvert$	$\lvert M\rvert$
2	3	3	17
3	4	6	14
4	6	13	7
5	8	20	3
6	9	21	3
7	10	29	3
8	15	36	0
9	19	45	0

Table 6.2: **Yeast exosome: oligomers and associated statistics.** The yeast exosome contains 10 proteins, with Csl4 found in size 9 oligomers only. **(1st column)** Size of oligomers i.e. number of subunits **(2nd column)** Number of oligomers up to a given size **(3rd column)** size of the pool of contacts associated with the oligomers selected. Note also that for $s = 8$ and $s = 9$, the pool size is maximal, i.e. contains all possible pairs of proteins: for $s = 8 : \binom{9}{2} = 36$; for $s = 9 : \binom{10}{2} = 45$. **(4th column)** The number of missed contacts, as defined on Fig. 6.2.

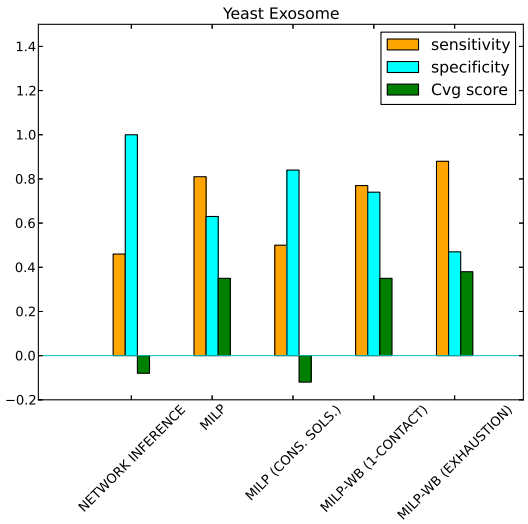


Figure 6.5: **Yeast Exosome: Assessment of contacts yielded from different algorithms, `MILP` and `MILP-W`$_{\mathbf{B}}$.** See supplemental Table 6.5 for the detailed statistics.

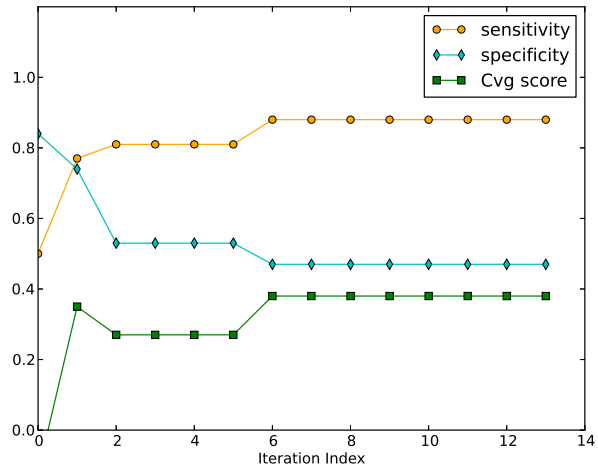


Figure 6.6: **Yeast Exosome: Variation of cumulative sensitivity, specificity and coverage score with iteration index.** Note that the iteration index also indicates number of contacts forbidden at a time. See supplemental Table 6.6 for the detailed statistics.

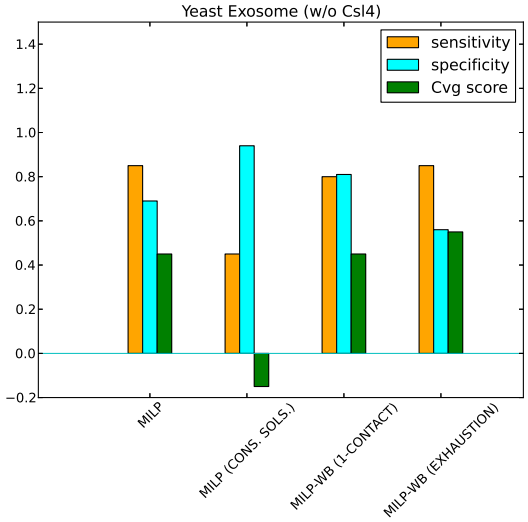


Figure 6.7: **Yeast Exosome without Csl4: Assessment of contacts yielded from different algorithms, `MILP` and `MILP-W`$_{\mathbf{B}}$.** See supplemental Table 6.5 for the detailed statistics.

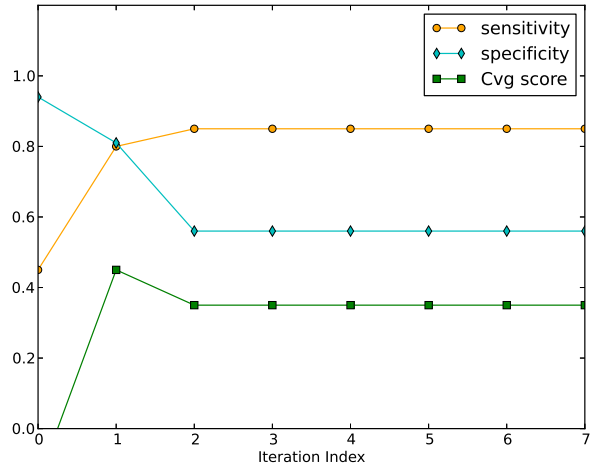


Figure 6.8: **Yeast Exosome without Csl4: Variation of cumulative sensitivity, specificity and coverage score with iteration index.** Note that the iteration index also indicates number of contacts forbidden at a time. See supplemental Table 6.7 for the detailed statistics.

6.7.3 Yeast Proteasome lid

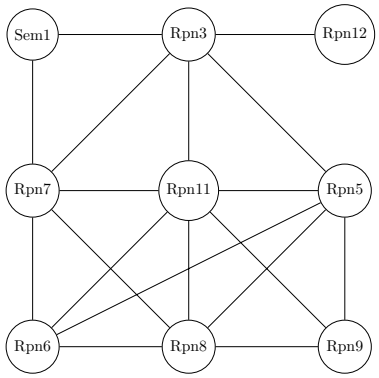


Figure 6.9: **Yeast proteasome lid: contacts between subunits.** Each edge corresponds to an interface between two subunits.

Oligomer size s	$\lvert\mathcal{O}_{\leq s}\rvert$	$\lvert\mathrm{Pool}_{\mathrm{E}}(\mathcal{O}_{\leq s})\rvert$	$\lvert M\rvert$
2	3	3	16
3	7	10	10
4	9	11	10
5	10	18	5
6	10	18	5
7	11	27	1
8	14	36	0

Table 6.3: **Proteasome lid: oligomers and associated statistics.** The yeast proteasome lid contains 9 proteins. Note that the pool size is maximal only for $s = 8$, the 14 oligomers yielding the 36 possible contacts. The value $s = 8$ also corresponds to a null number of missed contacts. See supplemental Table 6.2 for details on the notations.

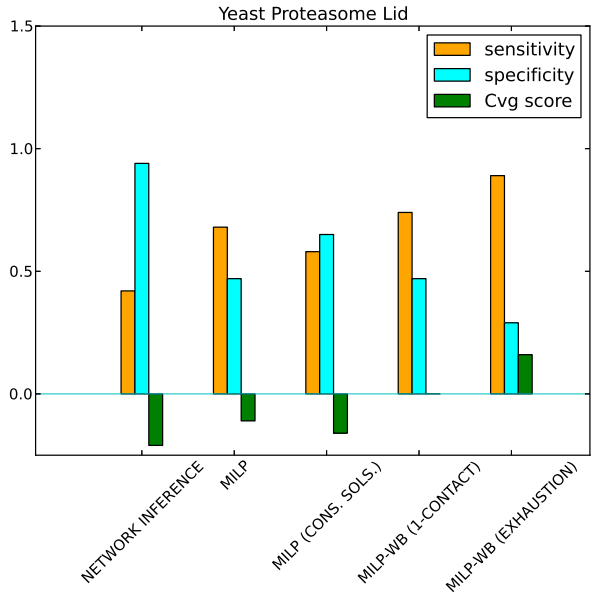


Figure 6.10: **Yeast Proteasome Lid: Assessment of contacts yielded from different algorithms, `MILP` and `MILP-W`$_{\mathbf{B}}$.** See supplemental Table 6.8 for the detailed statistics.

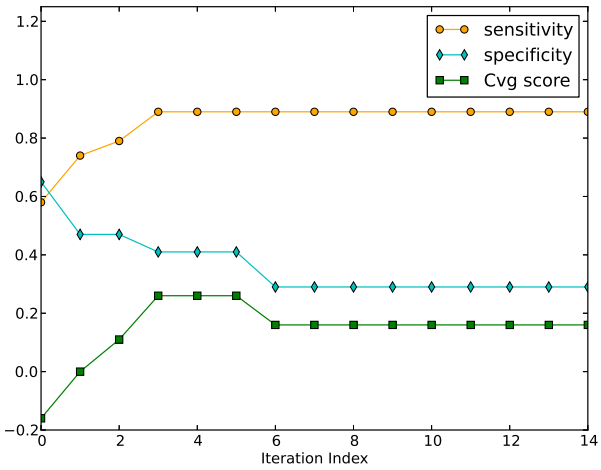


Figure 6.11: **Yeast Proteasome Lid: Variation of cumulative sensitivity, specificity and coverage score with iteration index.** Note that the iteration index also indicates number of contacts forbidden at a time. See supplemental Table 6.9 for the detailed statistics.

6.7.4 eIF3

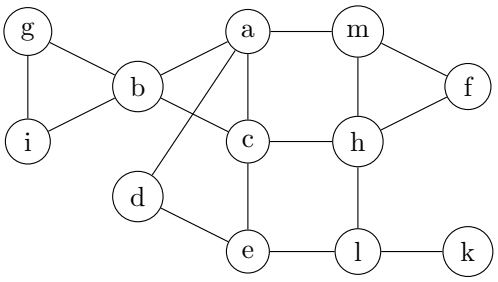


Figure 6.12: **Human eIF3: contacts between subunits.** Each edge corresponds to an interface between two subunits.

Oligomer size s	$\lvert\mathcal{O}_{\leq s}\rvert$	$\lvert\mathrm{Pool}_{\mathrm{E}}(\mathcal{O}_{\leq s})\rvert$	$\lvert M\rvert$
2	8	8	9
3	12	11	8
4	15	16	7
5	19	31	2
6	21	36	2
7	24	47	2
8	25	48	2
9	26	58	2
10	26	58	2
11	27	60	2

Table 6.4: **Human eIF3: oligomers and associated statistics.** The human eIF3 contains 12 proteins without eIF3j (a labile protein not present in the oligomers, see supplemental section 8.1). Note that the maximal pool size for 12 proteins is $\binom{12}{2} = 66$, however for $s = 11$, the pool size is 60, i.e. sub-maximal. The value $s = 11$ also lacks 2 reference contacts, i.e. $|M| = 2$ See Table 6.2 for details on the notations.

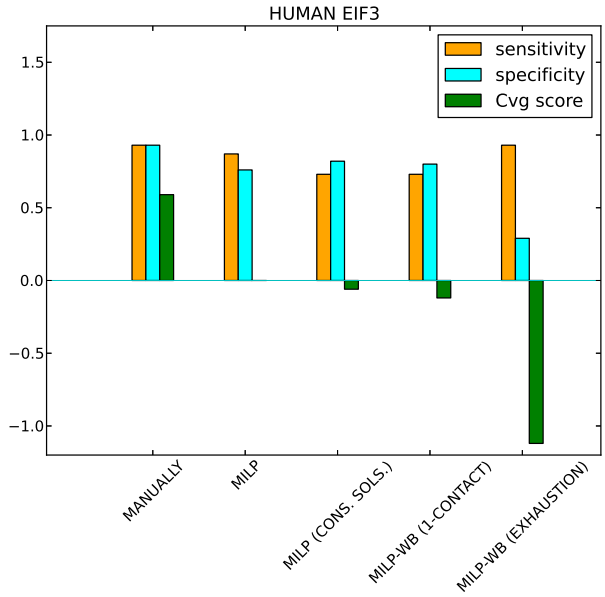


Figure 6.13: **Human eIF3: Assessment of contacts yielded from different algorithms, MILP and MILP-W_B.** See supplemental Table 6.10 for the detailed statistics.

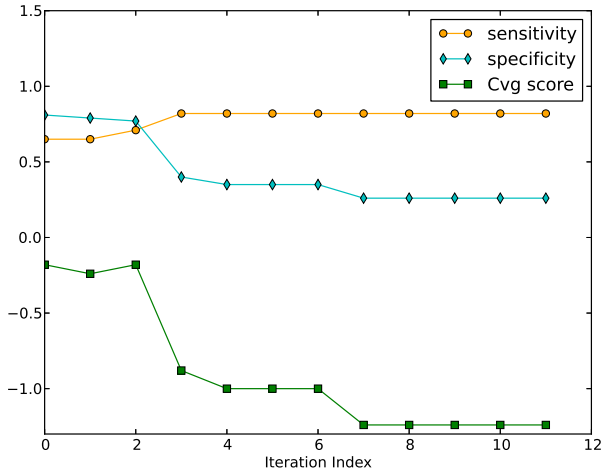


Figure 6.14: **Human eIF3: Variation of cumulative sensitivity, specificity and coverage score with iteration index.** Note that the iteration index also indicates number of contacts forbidden at a time. See supplemental Table 6.11 for the detailed statistics.

6.8 Supplemental: Results

6.8.1 Yeast Exosome

Results without Csl4. On solving the problem for yeast exosome (without Csl4) using `MILP` (or, `MILP-W` with $\alpha = 1$), one gets 10 consensus contacts in 2 consensus solutions (9 TP and 1 FP) (line with tag T6 in Table 6.5, and line with $n = 0$ in Table 6.6). We aim to enrich this initial set of consensus contacts, $\mathcal{E}^{cons.}_{\text{MILP}}$. Among these 10 contacts, we excluded 3 dimers in the set of oligomers (irreplaceable contacts), since, *ipso facto*, they are part of all the solutions, to launch the bootstrap procedure. The contacts from $\mathcal{E}^{cons.}_{\text{MILP}}$ are forbidden one at time by putting the label *forbidden* ('F'), yielding 7 different `MILP` problems each with different pool set (each having $|\text{Pool}_{\text{E}}(\mathcal{O}_{\leq s})| = 35$). The union of consensus contacts from 7 runs (including initial consensus contacts) has 19 contacts having 3 FP. The ROC scores are $\text{ROC}_{sens.}$ of 0.80, $\text{ROC}_{spec.}$ of 0.81 and $Cvg.$ score of 0.45 (T7 in Table 6.5 and row with $n = 1$ in Table 6.6).

If one goes further by forbidding two contacts at a time, there are 21 possible `MILP` problems each with different pool set (each having $|\text{Pool}_{\text{E}}(\mathcal{O}_{\leq s})| = 34$). The union of consensus contacts from these problems has 24 contacts having 7 FP. The cumulative number of contacts on taking the union from the first step of forbidding one contact at a time is also 24. Therefore, ROC scores are – $\text{ROC}_{sens.}$ of 0.85, $\text{ROC}_{spec.}$ of 0.56 and $Cvg.$ score of 0.35 ($n = 2$ in Table 6.6). These scores remain unchanged when we forbid 3,4,5 contacts at a time. When 6 or 7 contacts are forbidden, there is no solution since the pool set is insufficient ($n = 3, \ldots, 7$ in Table 6.6). Therefore, the final cumulative ROC score when all possible combinations of the contacts are precluded is $\text{ROC}_{sens.}$ of 0.85, $\text{ROC}_{spec.}$ of 0.56 and $Cvg.$ score of 0.35 (T8 in Table 6.5).

Results with Csl4. The complete system involves 10 proteins and 19 set of oligomers. The initial consensus set has 13 TP and 3 FP (T2 in Table 6.6) out of which 3 are dimers (irreplaceable contacts). On forbidding one contact at a time union of consensus contacts from 13 different `MILP` runs is 25 contacts with 20 TP and 5 FP. The corresponding ROC scores are i.e. $\text{ROC}_{sens.}$ of 0.77, $\text{ROC}_{spec.}$ of 0.74 and $Cvg.$ score of 0.35 (T3 in Table 6.5).

On forbidding further upto 11 contacts, the union of consensus contacts is 33 with 23 TP and 10 FP. Forbidding 12 and 13 contacts do not yield any solutions due to insufficient number of contacts. The ROC scores at the end, therefore, are $\text{ROC}_{sens.}$ of 0.88, $\text{ROC}_{spec.}$ of 0.47 and $Cvg.$ score of 0.38 (T4 in Table 6.5).

Assessment. Precluding the initial consensus contacts simultaneously yields new consensus contacts. We observe that the bootstrapping procedure has served its purpose which is to enrich the initial consensus contacts, $\mathcal{E}^{cons.}_{\text{MILP}}$ by possibly further sampling more TP and less FP. For $s = 8$, number of TP increases from 9 to 16 with 2 additional FP (T6 and T7 in Table 6.5). Similarly, for $s = 9$, TP increases from 13 to 20 with additional 2 FP (T2 and T3 in Table 6.5).

We also see that the bootstrapping procedure, $\texttt{MILP-W}_{\text{B}}$, which essentially extends the initial consensus contact set by including consensus contacts (high scoring contacts with high specificity) that are found in the way, in the end have comparable sensitivity and improved sensitivity to that of the contact sets yielded by `MILP`. For $s = 8$, for $\mathcal{E}_{\text{MILP}}$ and $\mathcal{E}_{\text{MILP-W}_{\text{B}}}$, respectively, the sensitivities are 0.85 and 0.80, whereas, the specificities are 0.69 and 0.81 (T5 vs T7 in Table 6.5). These numbers for $s = 9$ are 0.81 and 0.77 and 0.63 and 0.74 (T1 vs T3 in Table 6.5).

Note, that precluding more number of contacts though increase sensitivity as more TP are sampled but hurt the specificity as well as a result of sampling of FP (T4 and T8 in Table 6.5). This is due to rewiring of the system to a larger extent on forbidding large number of consensus contacts.

Finally, the performances are excellent when compared against those of the heuristic network algorithm [THS+08]. On the yeast exosome with Csl4, the sensitivity of $\texttt{MILP-W}_{\text{B}}$ is ~ 1.67 times that of network algorithm and $Cvg.$ score increases from -0.08 to 0.35 (T3 vs T0 in Table 6.5).

Tag	algo	s	$\lvert\text{Pool}_{\text{E}}(\mathcal{O}_{\leq s})\rvert$	M	P	TP	FN	N	TN	FP	$\text{ROC}_{sens.}$	$\text{ROC}_{spec.}$	Cvg
(T0)	*Network inference* [THS+08]	9	45	0	26	12	14	19	19	0	0.46	1	-0.08
(T1)	$\mathcal{E}_{\text{MILP}}$	9	45	0	26	21	5	19	12	7	0.81	0.63	0.35
(T2)	$\mathcal{E}^{cons.}_{\text{MILP}}$	9	45	0	26	13	13	19	16	3	0.50	0.84	-0.12
(T3)	$\mathcal{E}_{\text{MILP-W}_\text{B}}$, 1-contact	9	45	0	26	20	6	19	14	5	0.77	0.74	0.35
(T4)	$\mathcal{E}_{\text{MILP-W}_\text{B}}$, after exhaustion	9	45	0	26	23	3	19	9	10	0.88	0.47	0.38
(T5)	$\mathcal{E}_{\text{MILP}}$	8	36	0	20	17	3	16	11	5	0.85	0.69	0.45
(T6)	$\mathcal{E}^{cons.}_{\text{MILP}}$	8	36	0	20	9	11	16	15	1	0.45	0.94	-0.15
(T7)	$\mathcal{E}_{\text{MILP-W}_\text{B}}$, 1-contact	8	36	0	20	16	4	16	13	3	0.80	0.81	0.45
(T8)	$\mathcal{E}_{\text{MILP-W}_\text{B}}$, after exhaustion	8	36	0	20	17	3	16	9	7	0.85	0.56	0.35

Table 6.5: **Yeast exosome: sensitivity, specificity and coverage for various edge sets generated by `MILP` and `MILP-W`$_\mathbf{B}$.** Results from T0-T4 corresponds to Yeast exosome including Csl4 and from T5-T8 corresponds to Yeast exosome without Csl4. For a given run (each line), all edges predicted get distributed into TP and FP. In the following paragraph, the content in the bracket correspond to yeast exosome without Csl4. Out of a pool of candidate edges of size 45 (36), the edge set $\mathcal{E}_{\text{MILP-W}_\text{B}}$ (1st iteration) contains all true positives but six (but four), and five false positives (three false positives). It is to be noted that tag T3 (T7) corresponds to the 1st iteration of the Table 6.6 (Table 6.7), while tag T4 (T8) corresponds to results obtained upon precluding all possible combinations of initial consensus contacts.

#contacts fobidden, n	#combinations, $(\binom{13}{n})$	$\lvert\mathcal{E}^{cons.}_{\text{MILP-W}_\text{B}}\rvert$	$\lvert\mathcal{E}^{cons.}_{\text{MILP-W}_\text{B}}\rvert^{Cum.}$	individual			cumulative		
				$\text{ROC}_{sens.}$	$\text{ROC}_{spec.}$	Cvg	$\text{ROC}_{sens.}$	$\text{ROC}_{spec.}$	Cvg
0	1	16	16	0.50	0.84	-0.12	0.50	0.84	-0.12
1	13	25	25	0.77	0.74	0.35	0.77	0.74	0.35
2	78	30	30	0.81	0.53	0.27	0.81	0.53	0.27
3	286	30	30	0.81	0.53	0.27	0.81	0.53	0.27
4	715	30	30	0.81	0.53	0.27	0.81	0.53	0.27
5	1287	30	30	0.81	0.53	0.27	0.81	0.53	0.27
6	1716	33	33	0.88	0.47	0.38	0.88	0.47	0.38
7	1716	33	33	0.88	0.47	0.38	0.88	0.47	0.38
8	1287	33	33	0.88	0.47	0.38	0.88	0.47	0.38
9	715	33	33	0.88	0.47	0.38	0.88	0.47	0.38
10	286	33	33	0.88	0.47	0.38	0.88	0.47	0.38
11	78	25	33	0.77	0.74	0.35	0.88	0.47	0.38
12	13	0	33	-	-	-	0.88	0.47	0.38
13	1	0	24	-	-	-	0.88	0.47	0.38

Table 6.6: **Yeast exosome: sensitivity, specificity and coverage of enriched consensus set on forbidding a number of initial consensus contacts by `MILP-W`$_\mathbf{B}$.** Note that the cumulative statistics for row n is computed by considering union of all the consensus edge sets, $\mathcal{E}^{cons.}_{\text{MILP-W}_\text{B}}$ from 0 to $n = 13$.

#contacts fobidden, n	#combinations, $(\binom{7}{n})$	$\lvert\mathcal{E}^{cons.}_{\text{MILP-W}_\text{B}}\rvert$	$\lvert\mathcal{E}^{cons.}_{\text{MILP-W}_\text{B}}\rvert^{Cum.}$	individual			cumulative		
				ROC$_{sens.}$	ROC$_{spec.}$	Cvg	ROC$_{sens.}$	ROC$_{spec.}$	Cvg
0	1	10	10	0.45	0.94	-0.15	0.45	0.94	-0.15
1	7	19	19	0.80	0.81	0.45	0.80	0.81	0.45
2	21	24	24	0.85	0.56	0.35	0.85	0.56	0.35
3	35	24	24	0.85	0.56	0.35	0.85	0.56	0.35
4	35	24	24	0.85	0.56	0.35	0.85	0.56	0.35
5	21	24	24	0.85	0.56	0.35	0.85	0.56	0.35
6	7	0	24	-	-	-	0.85	0.56	0.35
7	1	0	24	-	-	-	0.85	0.56	0.35

Table 6.7: **Yeast exosome without Csl4: sensitivity, specificity and coverage of enriched consensus set on forbidding a number of initial consensus contacts by `MILP-W`$_\mathbf{B}$.** Note that the cumulative statistics for row n is computed by considering union of all the consensus edge sets, $\mathcal{E}^{cons.}_{\text{MILP-W}_\text{B}}$ from 0 to n.

6.8.2 Yeast 19S Proteasome lid

Results. This system involves 9 proteins and 14 oligomers. The initial consensus set using `MILP` (or `MILP-W` with $\alpha = 1$) has 11 TP and 6 FP out of which 3 are dimers (irreplaceable contacts) (line with tag T2 of Table 6.8, line with $n = 0$ in Table 6.9). On forbidding the contacts, $\mathcal{E}^{cons.}_{\text{MILP}}$ one at a time, one has 14 different `MILP` problems each with different pool set of size of 35. The union of consensus contacts of all such problems has 23 contacts having 14 TP and 9 FP. The ROC scores, therefore, are ROC$_{sens.}$ of 0.74, ROC$_{spec.}$ of 0.47 and $Cvg.$ score of 0 (T3 of the Table 6.8 and row with $n = 1$ of the Table 6.9). Proceeding further by precluding two contacts at a time, the size of union of consensus contacts yielded is 24 and cumulative union size (taking into account the previous step) is also 24. The ROC scores are - ROC$_{sens.}$ of 0.79, ROC$_{spec.}$ of 0.47 and $Cvg.$ score of 0.11 ($n = 2$ in Table 6.9). When three to five contacts are precluded then the following cumulative scores for 27 contacts are – ROC$_{sens.}$ of 0.89, ROC$_{spec.}$ of 0.41 and $Cvg.$ score of 0.26 ($n = 3, \ldots, 5$ in Table 6.9). When six contacts are precluded more FP are induced yielding scores – ROC$_{sens.}$ of 0.89, ROC$_{spec.}$ of 0.29 and $Cvg.$ score of 0.16. Beyond this point the cumulative scores do not change when number of contacts are precluded from 4 to 14 at a time ($n = 6, \ldots, 14$ in Table 6.9).

Assessment. The bootstrapping algorithm `MILP-W`$_\text{B}$ enriched the initial consensus contacts $\mathcal{E}^{cons.}_{\text{MILP}}$ by augmenting TP from 11 to 14 with 3 additional FP (T2 and T3 of the Table 6.8). On comparing with $\mathcal{E}_{\text{MILP}}$, we observe that for $\mathcal{E}_{\text{MILP-W}_\text{B}}$ the number of FP are same but number of TP is one more, thus improved sensitivity.

We again observe that on precluding more consensus contacts at a time, the specificity is dropped. The final cumulative score is ROC$_{sens.}$ of 0.89, ROC$_{spec.}$ of 0.29 and $Cvg.$ score of 0.16 (T4 of the Table 6.8 and Table 6.9).

Finally, when we compare with previously published contacts by *Network inference* algorithm in [THS$^+$08], we observe that the sensitivity for $\mathcal{E}_{\text{MILP-W}_\text{B}}$ is 1.76 higher than those published earlier. Also, $Cvg.$ score increases from -0.21 to 0 (T3 vs T0 in the Table 6.8).

Tag	algo	s	$\lvert\text{Pool}_{\text{E}}(\mathcal{O}_{\leq s})\rvert$	M	P	TP	FN	N	TN	FP	$\text{ROC}_{sens.}$	$\text{ROC}_{spec.}$	Cvg
(T0)	*Network inference* [THS+08]	8	36	0	19	8	11	17	16	1	0.42	0.94	-0.21
(T1)	$\mathcal{E}_{\text{MILP}}$	8	36	0	19	13	6	17	8	9	0.68	0.47	-0.11
(T2)	$\mathcal{E}^{cons.}_{\text{MILP}}$	8	36	0	19	11	8	17	11	6	0.58	0.65	-0.16
(T3)	$\mathcal{E}_{\text{MILP-W}_\text{B}}$, 1-contact	8	36	0	19	14	5	17	8	9	0.74	0.47	0
(T4)	$\mathcal{E}_{\text{MILP-W}_\text{B}}$, After exhaustion	8	36	0	19	17	2	17	5	12	0.89	0.29	0.16

Table 6.8: **Yeast proteasome lid: sensitivity, specificity and coverage for various edge sets generated by `MILP` and `MILP-W`$_\text{B}$.** For a given run (each line), all edges predicted get distributed into TP and FP. Out of a pool of candidate edges of size 36, the edge set $\mathcal{E}_{\text{MILP-W}_\text{B}}$ (1st iteration) contains all true positive but five, and nine false positives. It is to be noted that tag T3 corresponds to the 1st iteration of the Table 6.9, while tag T4 corresponds to results obtained upon precluding all possible combinations of initial consensus contacts.

#contacts fobidden, n	#combinations, $(\binom{14}{n})$	$\lvert\mathcal{E}^{cons.}_{\text{MILP-W}_\text{B}}\rvert$	$\lvert\mathcal{E}^{cons.}_{\text{MILP-W}_\text{B}}\rvert^{Cum.}$	individual			cumulative		
				$\text{ROC}_{sens.}$	$\text{ROC}_{spec.}$	Cvg	$\text{ROC}_{sens.}$	$\text{ROC}_{spec.}$	Cvg
0	1	17	17	0.58	0.65	-0.16	0.58	0.65	-0.16
1	14	23	23	0.74	0.47	0	0.74	0.47	0
2	91	24	24	0.79	0.47	0.11	0.79	0.47	0.11
3	364	27	27	0.89	0.41	0.26	0.89	0.41	0.26
4	1001	27	27	0.89	0.41	0.26	0.89	0.41	0.26
5	2002	27	27	0.89	0.41	0.26	0.89	0.41	0.26
6	3003	29	29	0.89	0.29	0.16	0.89	0.29	0.16
7	3432	29	29	0.89	0.29	0.16	0.89	0.29	0.16
8	3003	29	29	0.89	0.29	0.16	0.89	0.29	0.16
9	2002	29	29	0.89	0.29	0.16	0.89	0.29	0.16
10	1001	29	29	0.89	0.29	0.16	0.89	0.29	0.16
11	364	29	29	0.89	0.29	0.16	0.89	0.29	0.16
12	91	28	29	0.84	0.29	0.05	0.89	0.29	0.16
13	14	0	29	-	-	-	0.89	0.29	0.16
14	1	0	29	-	-	-	0.89	0.29	0.16

Table 6.9: **Yeast proteasome assembly: sensitivity, specificity and coverage of enriched consensus set on forbidding a number of initial consensus contacts by `MILP-W`$_\text{B}$.** Note that the cumulative statistics for row n is computed by considering union of all the consensus edge sets, $\mathcal{E}^{cons.}_{\text{MILP-W}_\text{B}}$ from 0 to n.

6.8.3 Eukaryotic Translation factor eIF3

Results and Assessment. Regarding the reference contacts for human eIF3 we have cryo-EM reconstruction and MS, MS/MS dimers. We do not have cross-linking contacts for human eIF3. Therefore, the set of reference contacts is possibly not exhaustive. Also, pool set of contacts is sub-maximal since the size is 60 instead of 66 (for 12 vertices) and maximum 15 out of 17 positives could be sampled from the pool set (Table 6.4).

However, the behavior of $\mathcal{E}_{\text{MILP-W}_\text{B}}$ viz-a-viz $\mathcal{E}^{cons.}_{\text{MILP}}$ and $\mathcal{E}_{\text{MILP}}$ resembles that of the previous two systems. Using bootstrapping procedure we report 20 contacts with 11 TP and 9 FP. The contact set $\mathcal{E}_{\text{MILP-W}_\text{B}}$ has one more additional FP than that of $\mathcal{E}^{cons.}_{\text{MILP}}$ (T3 vs T2 in the Table 6.10). However, the specificity is still better than that of $\mathcal{E}_{\text{MILP}}$ (T3 vs T1 in the Table 6.10). We observe that precluding more than 1 contact at a time in `MILP-W`$_\text{B}$ yields low specificity (T3 vs T4 in the Table 6.10 and Table 6.11).

The previously published contacts in [ZSF+08] are computed manually using experimental information from various other sources (T0 in the Table 6.10).

Tag	algo	s	$\lvert\text{Pool}_{\text{E}}(\mathcal{O}_{\leq s})\rvert$	M	P	TP	FN	N	TN	FP	$\text{ROC}_{sens.}$	$\text{ROC}_{spec.}$	Cvg
(T0)	*Manually* [ZSF+08]	11	60	2	15	14	1	45	42	3	0.93	0.93	0.59
(T1)	$\mathcal{E}_{\text{MILP}}$	11	60	2	15	13	2	45	34	11	0.87	0.76	0
(T2)	$\mathcal{E}^{cons.}_{\text{MILP}}$	11	60	2	15	11	4	45	37	8	0.73	0.82	-0.06
(T3)	$\mathcal{E}_{\text{MILP-W}_\text{B}}$, 1-contact	11	60	2	15	11	4	45	36	9	0.73	0.80	-0.12
(T4)	$\mathcal{E}_{\text{MILP-W}_\text{B}}$, After exhaustion	11	60	2	15	14	1	45	13	32	0.93	0.29	-1.12

Table 6.10: **Sensitivity, specificity and coverage for various edge sets generated by `MILP` and `MILP-W`$_\mathbf{B}$ for human eIF3 assembly.** For a given run (each line), all edges predicted get distributed into TP and FP. Out of a pool of candidate edges of size 60, the edge set $\mathcal{E}_{\text{MILP-W}_\text{B}}$ (1st iteration) contains all true positive but four, and nine false positives. It is to be noted that tag T3 corresponds to the 1st iteration of the Table 6.11, while tag T4 corresponds to results obtained upon precluding all possible combinations of initial consensus contacts.

#contacts fobidden, n	#combinations, $(\binom{11}{n})$	$\lvert\mathcal{E}^{cons.}_{\text{MILP-W}_\text{B}}\rvert$	$\lvert\mathcal{E}^{cons.}_{\text{MILP-W}_\text{B}}\rvert^{Cum.}$	individual			cumulative		
				$\text{ROC}_{sens.}$	$\text{ROC}_{spec.}$	Cvg	$\text{ROC}_{sens.}$	$\text{ROC}_{spec.}$	Cvg
0	1	19	19	0.65	0.81	-0.18	0.65	0.81	-0.18
1	11	20	20	0.65	0.79	-0.24	0.65	0.79	-0.24
2	55	22	22	0.71	0.77	-0.18	0.71	0.77	-0.18
3	165	40	40	0.82	0.40	-0.88	0.82	0.40	-0.88
4	330	42	42	0.82	0.35	-1.00	0.82	0.35	-1.00
5	462	42	42	0.82	0.35	-1.00	0.82	0.35	-1.00
6	462	42	42	0.82	0.35	-1.00	0.82	0.35	-1.00
7	330	46	46	0.82	0.26	-1.24	0.82	0.26	-1.24
8	165	46	46	0.82	0.26	-1.24	0.82	0.26	-1.24
9	55	46	46	0.82	0.26	-1.24	0.82	0.26	-1.24
10	11	46	46	0.82	0.26	-1.24	0.82	0.26	-1.24
11	1	46	46	0.82	0.26	-1.24	0.82	0.26	-1.24

Table 6.11: **Sensitivity, specificity and coverage of enriched consensus set on forbidding a number of initial consensus contacts by `MILP-W`$_\mathbf{B}$ for human eIF3 assembly.** Note that the cumulative statistics for row n is computed by considering union of all the consensus edge sets, $\mathcal{E}^{cons.}_{\text{MILP-W}_\text{B}}$ from 0 to n.

6.8.4 Using Weights: an Illustration

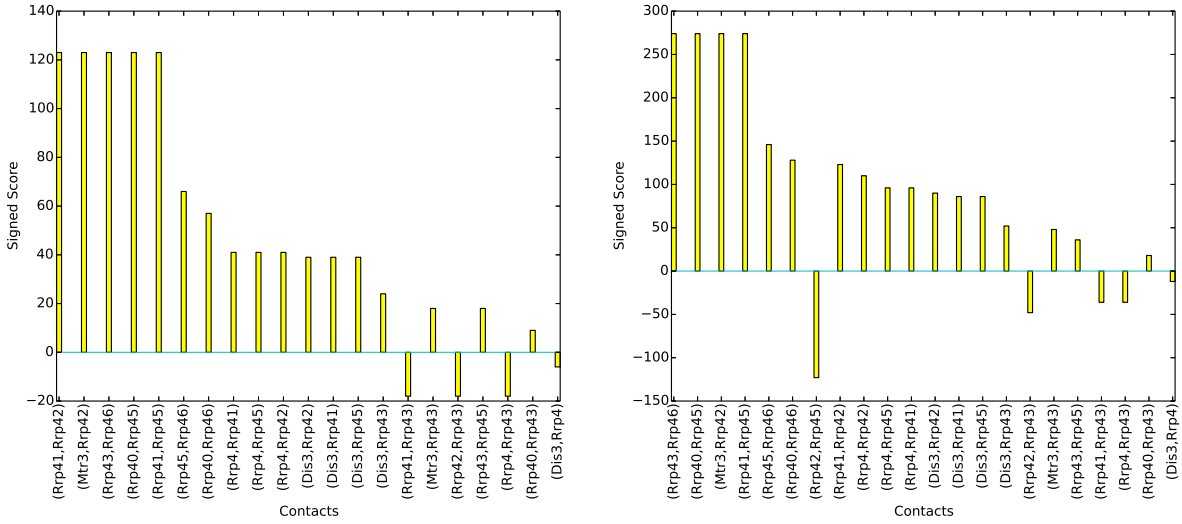


Figure 6.15: **Yeast Exosome: Contact distribution with [Left]** $\alpha = 0.25$ **and [Right]** $\alpha = 1.0$**.** Note that the contacts (Rrp45, Rrp46), (Rrp40, Rrp46) and (Rrp41, Rrp42) are assigned weights of 0.6 and rest are left at default (0.5).

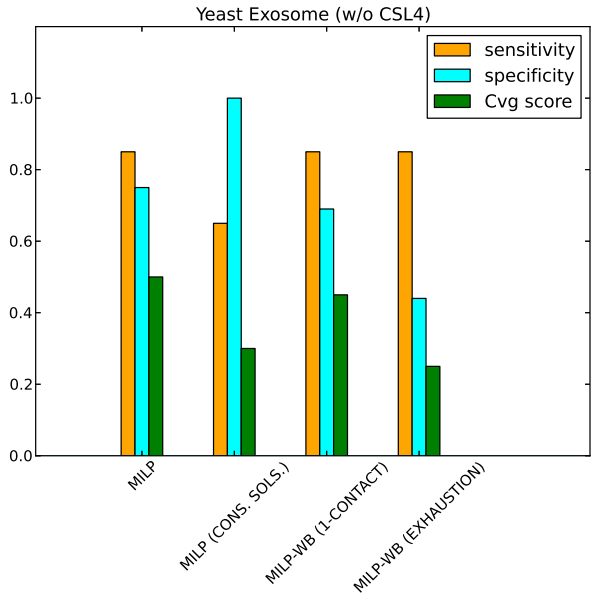


Figure 6.16: **Yeast Exosome (without Csl4): Assessment of contacts yielded from different algorithms, MILP and MILP-W_B with $\alpha = 0.25$.** See supplemental Table 6.12 for the detailed statistics. Note that the contacts (Rrp45, Rrp46), (Rrp40, Rrp46) and (Rrp41, Rrp42) are assigned weights of 0.6 and rest are left at default (0.5).

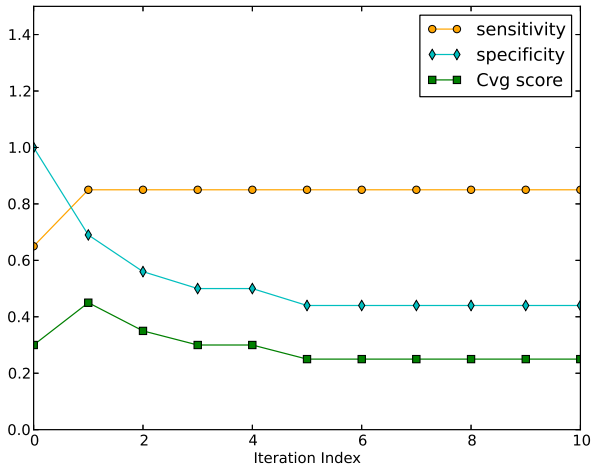


Figure 6.17: **Yeast Exosome (without Csl4): Evolution of cumulative sensitivity, specificity and coverage score with iteration index; $\alpha = 0.25$.** Note that the iteration index also indicates number of contacts forbidden at a time. See supplemental Table 6.13 for the detailed statistics. Note that the contacts (Rrp45, Rrp46), (Rrp40, Rrp46) and (Rrp41, Rrp42) are assigned weights of 0.6 and rest are left at default (0.5).

Tag	algo	s	$\lvert\text{Pool}_\text{E}(\mathcal{O}_{\leq s})\rvert$	M	P	TP	FN	N	TN	FP	$\text{ROC}_{sens.}$	$\text{ROC}_{spec.}$	Cvg
(T1)	$\mathcal{E}_{\text{MILP}}$	8	36	0	20	17	3	16	12	4	0.85	0.75	0.5
(T2)	$\mathcal{E}^{cons.}_{\text{MILP}}$	8	36	0	20	13	7	16	16	0	0.65	1.0	0.3
(T3)	$\mathcal{E}_{\text{MILP-W}_\text{B}}$, 1-contact	8	36	0	20	17	3	16	11	5	0.85	0.69	0.45
(T4)	$\mathcal{E}_{\text{MILP-W}_\text{B}}$, After exhaustion	8	36	0	20	17	3	16	7	9	0.85	0.44	0.25

Table 6.12: **Sensitivity, specificity and coverage for various edge sets generated by MILP and MILP-W_B for yeast exosome (without Csl4) assembly; $\alpha = 0.25$.** Note that the contacts (Rrp45, Rrp46), (Rrp40, Rrp46) and (Rrp41, Rrp42) are assigned weights of 0.6 and rest are left at default (0.5). For a given run (each line), all edges predicted get distributed into TP and FP. Out of a pool of candidate edges of size 36, the edge set $\mathcal{E}_{\text{MILP-W}_\text{B}}$ (1st iteration) contains all true positive but three, and five false positives. It is to be noted that tag T3 corresponds to the 1st iteration of the Table 6.13, while tag T4 corresponds to results obtained upon precluding all possible combinations of initial consensus contacts.

#contacts fobidden, n	#combinations, $(\binom{10}{n})$	$\lvert\mathcal{E}^{cons.}_{\text{MILP-W}_\text{B}}\rvert$	$\lvert\mathcal{E}^{cons.}_{\text{MILP-W}_\text{B}}\rvert^{Cum.}$	individual			cumulative		
				$\text{ROC}_{sens.}$	$\text{ROC}_{spec.}$	Cvg	$\text{ROC}_{sens.}$	$\text{ROC}_{spec.}$	Cvg
0	1	13	13	0.65	1	0.3	0.65	1	0.3
1	10	22	22	0.85	0.69	0.45	0.85	0.69	0.45
2	45	24	24	0.85	0.56	0.35	0.85	0.56	0.35
3	120	25	25	0.85	0.50	0.30	0.85	0.50	0.30
4	210	23	25	0.85	0.63	0.40	0.85	0.50	0.30
5	252	24	26	0.85	0.56	0.35	0.85	0.44	0.25
6	210	24	26	0.85	0.56	0.35	0.85	0.44	0.25
7	120	24	26	0.85	0.56	0.35	0.85	0.44	0.25
8	45	24	26	0.85	0.56	0.35	0.85	0.44	0.25
9	10	-	-	-	-	-	0.85	0.44	0.25
10	1	-	-	-	-	-	0.85	0.44	0.25

Table 6.13: **Yeast exosome (without Csl4): sensitivity, specificity and coverage of enriched consensus set on forbidding a number of initial consensus contacts by `MILP-W`$_\mathbf{B}$; $\alpha = 0.25$.** The contacts (Rrp45, Rrp46), (Rrp40, Rrp46) and (Rrp41, Rrp42) are assigned weights of 0.6 and rest are left at default (0.5). Note that the cumulative statistics for row n is computed by considering union of all the consensus edge sets, $\mathcal{E}^{cons.}_{\text{MILP-W}_\text{B}}$ from 0 to n.

6.9 Supplemental: Algorithms and Programs

6.9.1 Problem hardness, existing algorithms and contributions

Assessing the intrinsic difficulty of a combinatorial problem requires inspecting the *decision* and the *optimization* versions of the problem [GJ79]. In our case, deciding whether a MCI problem admits a solution using a pre-defined number of edges k is **NP**-complete, while finding the solution of smallest size is APX-hard. This latter result is of special interest since we aim at finding an edge set of minimal size. It stipulates that unless $\mathbf{P} = \mathbf{NP}$, there does not exist any polynomial time approximation scheme [AAC+13], that is, a polynomial time algorithm reporting an edge set *as close as desired*, in terms of size, from the optimum. It should be stressed that these facts do not exploit any peculiar property of real data, and only show the existence of hard i.e. difficult to solve instances.

6.9.2 Algorithm MILP-$\mathrm{W_B}$: pseudo-code

Algorithm 3 Algorithm MILP-$\mathrm{W_B}$, with initial call MILP-$\mathrm{W_B}$ ($\mathcal{E}^{cons.}_{\mathbf{MILP}} \backslash I$), with I standing for the list of dimers in the list of oligomers. The algorithm bootstraps from consensus edges, and collects novel consensus edges which appear upon precluding already found consensus edges.

Require: $\mathcal{E}^{cons.}_{\mathrm{MILP}}$ – initial consensus edges.
Require: I – irreplaceable contacts (dimers).
Require: $spec_0$ – the initial connectivity inference specification.
Require: $\mathcal{E}_{\text{MILP-W}_\text{B}}$ – the set storing all consensus edges.
Parameter: B – consensus edges to be precluded $:= \mathcal{E}^{cons.}_{\mathrm{MILP}} \setminus I$

Algorithm: MILP-$\mathrm{W_B}$ (B)

```
1:  E_MILP-W_B ← E^cons._MILP
2:  /* NB: an iteration precludes all possible ℓ-tuples */
3:  for ℓ from 1 to |B| do
4:     Get fsets_ℓ: all ℓ-tuples from the set B, namely (B choose ℓ)
5:     /* C_ℓ: A set of consensus contacts that will be generated for all ℓ-tuples is initiated to E^cons._MILP */
6:     C_ℓ ← ∅
7:     for each fset ∈ fsets_ℓ do
8:        /* Edit the initial specification spec_0 to take into account the annotations */
9:        Assign label forbidden ('F') to all contacts in fset
10:       Run MILP-W for this novel specification
11:       Get consensus contacts associated with fset, denoted C_fset
12:       C_ℓ ← C_ℓ ∪ C_fset
13:    E_MILP-W_B = E_MILP-W_B ∪ C_ℓ
```

6.9.3 Implementation

Our algorithms have been implemented using IBM CPLEX solver 12.6. The typical running time required to solve an instance presented in this paper is circa 30 seconds, on a standard laptop computer (2.80GHz Intel(R) Xeon(R) CPU E5-1603 0).

Upon publication of this paper, the programs implementing MILP, MILP-W, and MILP-$\mathrm{W_B}$ will be distributed within the *Structural Bioinformatics Library* (http://structural-bioinformatics-library.org/).

6.10 Supplemental: Using Weights: a Detailed Study

6.10.1 Methods

In the sequel, we present a thorough evaluation of `MILP-W` upon varying weights and the value of α – Eq. (6.1).

To this end, we challenge algorithm `MILP-W` with two classes of instances. While *deterministic instances* are meant to assess the behavior of the algorithm under controlled conditions, *randomized instances* are meant to investigate scenarios where no a priori information on the contacts is known.

Deterministic instances

Specification. The input specification of a MWCI problem depends to three ingredients, namely the set of oligomers $\mathcal{O}_{\leq s}$, the value of α, and the individual weights $w(\cdot)$ for the candidate edges in $\text{Pool}_{\text{E}}(\mathcal{O}_{\leq s})$. We design MWCI instances to assess the relative importance of these ingredients. To this end, consider two values $F > G = 0.5 > U$, respectively meant to favor and penalize contacts. Note that the value $G = 0.5$ is a default value for contacts for which there is no a priori. The gap between these two values is defined by $\Delta = F - U$. Practically, we consider three cases, namely $(F, U) = (0.9, 0.1)$, $(F, U) = (0.75, 0.25)$, and $(F, U) = (0.6, 0.4)$,

The first set of instances involves the two weights F and U applied to the edges of the pool. The instance FU is obtained by assigning the weight F to all TP, and the weight U to all FP. To define a control, we define the UF instance by swapping the weights i.e. by favoring FP and penalizing TP. Note that instances of the type FF or UU, where true and false positives are given the same weight, are irrelevant since they are covered by the case $\alpha = 1$.

We first report basic facts observed for deterministic instances FU and UF defined by oligomers of size $s = 5, 8$, since the cases $s = 6, 7$ match $s = 5$ (supplemental Tables 6.14, 6.15, and 6.16).

Results. We examine successively the roles of α and of the individual weights.

Parameter α. When α increases, two striking facts are observed. First, the number of solutions increases, since one has up to 9 solutions when $\alpha = 0.25$, but up to 274 solutions when $\alpha = 1$ (supp. Table 6.14, $s = 8$). This solution set uses 22 contacts out of a pool of size 36. These 22 contacts involves 17 TP and 5 FP, resulting in a coverage of 0.45. The maximal number of solutions for $\alpha = 1$ owes to the fact that ties between contacts cannot be broken thanks to the weights, so that all solutions with the same number of contacts are equivalent. Second, the size of solutions decreases (up to 22 contacts for $\alpha = 0.25$ but nine only for $\alpha = 1$). This owes to the modest constant overhead in Eq. (6.2) for small values of α.

Weights. The configuration yielding the maximum number of solutions comes with an average (0.45) coverage (supp. Table 6.14, $s = 8$). Improving this score requires optimized combinations of α and weights, which is observed for the FU instance and $\alpha = 0.25$. In that case, the 20 TP are reported, while no FP is found, resulting in perfect unit values for the sensitivity, the specificity, and the coverage. This is admittedly a contrived experiments since TP are promoted while FP are hindered. Reverting odds, the control setup UF yields the expected, since penalizing TP and promoting FP results in a poor coverage (from one for FU to -0.90 for UF). It is also noticed that the difference in coverage decreases when α increases. For example, considering oligomers of size five, one gets $0.95(= 0.85 - (-0.1))$, $0.5(= 0.45 - (-0.05))$, and $0(= 0.35 - 0.35)$ for $\alpha = 0.25, 0.5, 1$ respectively (supp. Table 6.14, $s = 5$). This owes to the decreasing prevalence of weights when α increases. In a similar vein, larger values of Δ, or equivalently large values of the weight F favor high coverages (for the FU case, $\alpha = 0.25$ and $s = 8$, the coverage drops from one to 0.7 in moving from $\Delta = 0.8$ to $\Delta = 0.2$.)

All versus consensus solutions. Consensus solutions, which form a subset of all solutions, are characterized by two main properties. First, the number of consensus solutions varies in the range 1 to 48, that

is, one get a 6 fold reduction with respect to the max number of total solutions. Second, the number of solutions is accompanied by a smaller set of edges used out of the pool of size 36, and also a smaller number (often null) of false positives. The former number decreases faster than the later, whence, overall, lower coverages.

Randomized instances

Specification. In designing deterministic instances involving the weights F and U, some a priori knowledge on the individual contacts is required to favor contacts standing a better chance to be true positives. If such information is not available, one could use favorable or unfavorable weights only. However, from the analysis carried out on deterministic instances, one gets that the FF scenario yields large solutions with false positives, while the UU scenario yields poor statistics — and in the extreme case connectivity inference problems without any solution. We therefore design a new class of instances also involving the intermediate weight G.

To specify these instances, we start from a deterministic instance, and use randomization. Consider e.g. the assignment of weights TP $\leftrightarrow F$ and FP $\leftrightarrow U$. For each contact from FP, we toss a fair coin and proceed as follows: if head is obtained, the contact keeps the weight F; if not, its weight is changed to G. We proceed likewise for false positive contacts, which may then be re-assigned a weight of G instead of the initial weight U. Note that for a given set of contacts (TP or FP), the expectation of the number of contacts whose weight is changed is half of the size of that set since the coin is fair. To avoid random bias, we generate 20 such instances.

Results. We noticed above that the FF and UU cases in the deterministic setting actually correspond to the case $\alpha = 1$. In comparing the results for randomized FF and UU instances against the case $\alpha = 1$, one first notices a drastic decrease of the number of solutions (2 for FF and $\alpha = 0.25$, 7 for UU and $\alpha = 0.25$, versus 274 for $\alpha = 1$) (Table 6.17). Solution size, however, are coherent with the deterministic case, and depend on the weights (large solutions for F weights, small solutions for U weights). Most interesting is the analysis of UU instances. On the one hand, a satisfactory sensitivity is obtained (for $\alpha = 0.25$: $\mathrm{ROC}_{sens.} = 0.55$ for randomized instances, versus $\mathrm{ROC}_{sens.} = 0.85$ for deterministic instances). On the other hand, an excellent specificity is observed (for $\alpha = 0.25$: $\mathrm{ROC}_{spec.} = 0.91$ for randomized instances, versus $\mathrm{ROC}_{spec.} = 0.69$ for deterministic instances).

Overall recommendations

We summarize the insights gained from the previous experiments on deterministic and randomized instances:

(i) Low values of α are sensitive to weights on the edges, as large solutions arise from favored edges.

(ii) Consensus solutions strongly hint at contacts which are true positives. However, modest coverage may stem from many false negatives.

(iii) High coverage scores are observed in two cases, namely when large solutions are obtained, or when a large number of solutions are obtained.

(iv) The scenario consisting of hindering a fraction of true contacts (by unfavorable weights or removing them from the pool) may trigger the discovery of alternative contacts also satisfying the connectivity constraints of oligomers. This finding, which stems from the analysis of randomized instances, underlies the strategy used in Algorithm `MILP-W`$_{\mathrm{B}}$ (section 6.5.1).

6.10.2 Results

The following tables present statistics to assess the incidence of weights, as explained in the main text. The following comments are in order:

- In the tables, the coverage values of Eq. (6.5) are color coded with a heat map, from blue (0-0.1) to red (0.9 - 1).
- The values reported in Tables 6.17, 6.18, 6.19 were obtained on 20 runs. The statistics reported correspond to the median of the values. For example, the number of solutions and the solution size are the median of the values obtained for all runs.

					$\alpha = 0.25$		$\alpha = 0.50$		$\alpha = 0.75$		$\alpha = 1$	
oligomer size, s	$\lvert\text{Pool}_{\text{E}}(\mathcal{O}_{\leq s})\rvert$	$\lvert M\rvert$	mode	*sols type*	(ROC$_{sens.}$, ROC$_{spec.}$, Cvg)	Solutions (#, size)	(ROC$_{sens.}$, ROC$_{spec.}$, Cvg)	Solutions (#, size)	(ROC$_{sens.}$, ROC$_{spec.}$, Cvg)	Solutions (#, size)	(ROC$_{sens.}$, ROC$_{spec.}$, Cvg)	Solutions (#, size)
5	20	3	FU	all	(1.00, 1.00, 0.85)	(1, 17)	(0.76, 1.00, 0.45)	(45, 8)	(0.76, 1.00, 0.45)	(45, 8)	(0.76, 0.33, 0.35)	(96, 8)
5	20	3	FU	cons	(1.00, 1.00, 0.85)	(1, 17)	(0.65, 1.00, 0.25)	(9, 8)	(0.65, 1.00, 0.25)	(9, 8)	(0.76, 0.33, 0.35)	(48, 8)
5	20	3	UF	all	(0.53, 0.00, -0.10)	(9, 9)	(0.53, 0.33, -0.05)	(9, 8)	(0.53, 0.33, -0.05)	(9, 8)	(0.76, 0.33, 0.35)	(96, 8)
5	20	3	UF	cons	(0.53, 0.00, -0.10)	(9, 9)	(0.53, 0.33, -0.05)	(9, 8)	(0.53, 0.33, -0.05)	(9, 8)	(0.76, 0.33, 0.35)	(48, 8)
6	21	3	FU	all	(1.00, 1.00, 0.85)	(1, 17)	(0.76, 1.00, 0.45)	(45, 8)	(0.76, 1.00, 0.45)	(45, 8)	(0.76, 0.50, 0.35)	(96, 8)
6	21	3	FU	cons	(1.00, 1.00, 0.85)	(1, 17)	(0.65, 1.00, 0.25)	(9, 8)	(0.65, 1.00, 0.25)	(9, 8)	(0.76, 0.50, 0.35)	(48, 8)
6	21	3	UF	all	(0.53, 0.00, -0.15)	(9, 10)	(0.53, 0.50, -0.05)	(9, 8)	(0.53, 0.50, -0.05)	(9, 8)	(0.76, 0.50, 0.35)	(96, 8)
6	21	3	UF	cons	(0.53, 0.00, -0.15)	(9, 10)	(0.53, 0.50, -0.05)	(9, 8)	(0.53, 0.50, -0.05)	(9, 8)	(0.76, 0.50, 0.35)	(48, 8)
7	29	3	FU	all	(1.00, 1.00, 0.85)	(1, 17)	(0.76, 1.00, 0.45)	(45, 8)	(0.76, 1.00, 0.45)	(45, 8)	(0.76, 0.83, 0.35)	(96, 8)
7	29	3	FU	cons	(1.00, 1.00, 0.85)	(1, 17)	(0.65, 1.00, 0.25)	(9, 8)	(0.65, 1.00, 0.25)	(9, 8)	(0.76, 0.83, 0.35)	(48, 8)
7	29	3	UF	all	(0.53, 0.00, -0.55)	(9, 18)	(0.53, 0.83, -0.05)	(9, 8)	(0.53, 0.83, -0.05)	(9, 8)	(0.76, 0.83, 0.35)	(96, 8)
7	29	3	UF	cons	(0.53, 0.00, -0.55)	(9, 18)	(0.53, 0.83, -0.05)	(9, 8)	(0.53, 0.83, -0.05)	(9, 8)	(0.76, 0.83, 0.35)	(48, 8)
8	36	0	FU	all	(1.00, 1.00, 1.00)	(1, 20)	(0.85, 1.00, 0.70)	(79, 9)	(0.85, 1.00, 0.70)	(79, 9)	(0.85, 0.69, 0.45)	(274, 9)
8	36	0	FU	cons	(1.00, 1.00, 1.00)	(1, 20)	(0.45, 1.00, -0.10)	(1, 9)	(0.45, 1.00, -0.10)	(1, 9)	(0.45, 0.94, -0.15)	(2, 9)
8	36	0	UF	all	(0.45, 0.00, -0.90)	(9, 22)	(0.45, 0.50, -0.50)	(63, 10)	(0.65, 0.69, 0.05)	(60, 9)	(0.85, 0.69, 0.45)	(274, 9)
8	36	0	UF	cons	(0.45, 0.00, -0.90)	(9, 22)	(0.45, 0.63, -0.40)	(18, 10)	(0.60, 0.75, 0.00)	(54, 9)	(0.45, 0.94, -0.15)	(2, 9)

Table 6.14: **Yeast exosome: statistics for U=0.1, F=0.9.**

					$\alpha = 0.25$		$\alpha = 0.50$		$\alpha = 0.75$		$\alpha = 1$	
oligomer size, s	$\lvert\text{Pool}_{\text{E}}(\mathcal{O}_{\leq s})\rvert$	$\lvert M\rvert$	mode	*sols type*	(ROC$_{sens.}$, ROC$_{spec.}$, Cvg)	Solutions (#, size)	(ROC$_{sens.}$, ROC$_{spec.}$, Cvg)	Solutions (#, size)	(ROC$_{sens.}$, ROC$_{spec.}$, Cvg)	Solutions (#, size)	(ROC$_{sens.}$, ROC$_{spec.}$, Cvg)	Solutions (#, size)
5	20	3	FU	all	(0.76, 1.00, 0.45)	(45, 8)	(0.76, 1.00, 0.45)	(45, 8)	(0.76, 1.00, 0.45)	(45, 8)	(0.76, 0.33, 0.35)	(96, 8)
5	20	3	FU	cons	(0.65, 1.00, 0.25)	(9, 8)	(0.65, 1.00, 0.25)	(9, 8)	(0.65, 1.00, 0.25)	(9, 8)	(0.76, 0.33, 0.35)	(48, 8)
5	20	3	UF	all	(0.53, 0.33, -0.05)	(9, 8)	(0.53, 0.33, -0.05)	(9, 8)	(0.53, 0.33, -0.05)	(9, 8)	(0.76, 0.33, 0.35)	(96, 8)
5	20	3	UF	cons	(0.53, 0.33, -0.05)	(9, 8)	(0.53, 0.33, -0.05)	(9, 8)	(0.53, 0.33, -0.05)	(9, 8)	(0.76, 0.33, 0.35)	(48, 8)
6	21	3	FU	all	(0.76, 1.00, 0.45)	(45, 8)	(0.76, 1.00, 0.45)	(45, 8)	(0.76, 1.00, 0.45)	(45, 8)	(0.76, 0.50, 0.35)	(96, 8)
6	21	3	FU	cons	(0.65, 1.00, 0.25)	(9, 8)	(0.65, 1.00, 0.25)	(9, 8)	(0.65, 1.00, 0.25)	(9, 8)	(0.76, 0.50, 0.35)	(48, 8)
6	21	3	UF	all	(0.53, 0.50, -0.05)	(9, 8)	(0.53, 0.50, -0.05)	(9, 8)	(0.53, 0.50, -0.05)	(9, 8)	(0.76, 0.50, 0.35)	(96, 8)
6	21	3	UF	cons	(0.53, 0.50, -0.05)	(9, 8)	(0.53, 0.50, -0.05)	(9, 8)	(0.53, 0.50, -0.05)	(9, 8)	(0.76, 0.50, 0.35)	(48, 8)
7	29	3	FU	all	(0.76, 1.00, 0.45)	(45, 8)	(0.76, 1.00, 0.45)	(45, 8)	(0.76, 1.00, 0.45)	(45, 8)	(0.76, 0.83, 0.35)	(96, 8)
7	29	3	FU	cons	(0.65, 1.00, 0.25)	(9, 8)	(0.65, 1.00, 0.25)	(9, 8)	(0.65, 1.00, 0.25)	(9, 8)	(0.76, 0.83, 0.35)	(48, 8)
7	29	3	UF	all	(0.53, 0.83, -0.05)	(9, 8)	(0.53, 0.83, -0.05)	(9, 8)	(0.53, 0.83, -0.05)	(9, 8)	(0.76, 0.83, 0.35)	(96, 8)
7	29	3	UF	cons	(0.53, 0.83, -0.05)	(9, 8)	(0.53, 0.83, -0.05)	(9, 8)	(0.53, 0.83, -0.05)	(9, 8)	(0.76, 0.83, 0.35)	(48, 8)
8	36	0	FU	all	(0.85, 1.00, 0.70)	(79, 9)	(0.85, 1.00, 0.70)	(79, 9)	(0.85, 1.00, 0.70)	(79, 9)	(0.85, 0.69, 0.45)	(274, 9)
8	36	0	FU	cons	(0.45, 1.00, -0.10)	(1, 9)	(0.45, 1.00, -0.10)	(1, 9)	(0.45, 1.00, -0.10)	(1, 9)	(0.45, 0.94, -0.15)	(2, 9)
8	36	0	UF	all	(0.45, 0.50, -0.50)	(63, 10)	(0.65, 0.69, 0.05)	(60, 9)	(0.65, 0.69, 0.05)	(60, 9)	(0.85, 0.69, 0.45)	(274, 9)
8	36	0	UF	cons	(0.45, 0.63, -0.40)	(18, 10)	(0.60, 0.75, 0.00)	(54, 9)	(0.60, 0.75, 0.00)	(54, 9)	(0.45, 0.94, -0.15)	(2, 9)

Table 6.15: **Yeast exosome: statistics for U=0.25, F=0.75.**

					$\alpha = 0.25$		$\alpha = 0.50$		$\alpha = 0.75$		$\alpha = 1$	
oligomer size, s	$\lvert\text{Pool}_{\text{E}}(\mathcal{O}_{\leq s})\rvert$	$\lvert M\rvert$	mode	*sols type*	(ROC$_{sens.}$, ROC$_{spec.}$, Cvg)	Solutions (#, size)	(ROC$_{sens.}$, ROC$_{spec.}$, Cvg)	Solutions (#, size)	(ROC$_{sens.}$, ROC$_{spec.}$, Cvg)	Solutions (#, size)	(ROC$_{sens.}$, ROC$_{spec.}$, Cvg)	Solutions (#, size)
5	20	3	FU	all	(0.76, 1.00, 0.45)	(45, 8)	(0.76, 1.00, 0.45)	(45, 8)	(0.76, 1.00, 0.45)	(45, 8)	(0.76, 0.33, 0.35)	(96, 8)
5	20	3	FU	cons	(0.65, 1.00, 0.25)	(9, 8)	(0.65, 1.00, 0.25)	(9, 8)	(0.65, 1.00, 0.25)	(9, 8)	(0.76, 0.33, 0.35)	(48, 8)
5	20	3	UF	all	(0.53, 0.33, -0.05)	(9, 8)	(0.53, 0.33, -0.05)	(9, 8)	(0.53, 0.33, -0.05)	(9, 8)	(0.76, 0.33, 0.35)	(96, 8)
5	20	3	UF	cons	(0.53, 0.33, -0.05)	(9, 8)	(0.53, 0.33, -0.05)	(9, 8)	(0.53, 0.33, -0.05)	(9, 8)	(0.76, 0.33, 0.35)	(48, 8)
6	21	3	FU	all	(0.76, 1.00, 0.45)	(45, 8)	(0.76, 1.00, 0.45)	(45, 8)	(0.76, 1.00, 0.45)	(45, 8)	(0.76, 0.50, 0.35)	(96, 8)
6	21	3	FU	cons	(0.65, 1.00, 0.25)	(9, 8)	(0.65, 1.00, 0.25)	(9, 8)	(0.65, 1.00, 0.25)	(9, 8)	(0.76, 0.50, 0.35)	(48, 8)
6	21	3	UF	all	(0.53, 0.50, -0.05)	(9, 8)	(0.53, 0.50, -0.05)	(9, 8)	(0.53, 0.50, -0.05)	(9, 8)	(0.76, 0.50, 0.35)	(96, 8)
6	21	3	UF	cons	(0.53, 0.50, -0.05)	(9, 8)	(0.53, 0.50, -0.05)	(9, 8)	(0.53, 0.50, -0.05)	(9, 8)	(0.76, 0.50, 0.35)	(48, 8)
7	29	3	FU	all	(0.76, 1.00, 0.45)	(45, 8)	(0.76, 1.00, 0.45)	(45, 8)	(0.76, 1.00, 0.45)	(45, 8)	(0.76, 0.83, 0.35)	(96, 8)
7	29	3	FU	cons	(0.65, 1.00, 0.25)	(9, 8)	(0.65, 1.00, 0.25)	(9, 8)	(0.65, 1.00, 0.25)	(9, 8)	(0.76, 0.83, 0.35)	(48, 8)
7	29	3	UF	all	(0.53, 0.83, -0.05)	(9, 8)	(0.53, 0.83, -0.05)	(9, 8)	(0.53, 0.83, -0.05)	(9, 8)	(0.76, 0.83, 0.35)	(96, 8)
7	29	3	UF	cons	(0.53, 0.83, -0.05)	(9, 8)	(0.53, 0.83, -0.05)	(9, 8)	(0.53, 0.83, -0.05)	(9, 8)	(0.76, 0.83, 0.35)	(48, 8)
8	36	0	FU	all	(0.85, 1.00, 0.70)	(79, 9)	(0.85, 1.00, 0.70)	(79, 9)	(0.85, 1.00, 0.70)	(79, 9)	(0.85, 0.69, 0.45)	(274, 9)
8	36	0	FU	cons	(0.45, 1.00, -0.10)	(1, 9)	(0.45, 1.00, -0.10)	(1, 9)	(0.45, 1.00, -0.10)	(1, 9)	(0.45, 0.94, -0.15)	(2, 9)
8	36	0	UF	all	(0.65, 0.69, 0.05)	(60, 9)	(0.65, 0.69, 0.05)	(60, 9)	(0.65, 0.69, 0.05)	(60, 9)	(0.85, 0.69, 0.45)	(274, 9)
8	36	0	UF	cons	(0.60, 0.75, 0.00)	(54, 9)	(0.60, 0.75, 0.00)	(54, 9)	(0.60, 0.75, 0.00)	(54, 9)	(0.45, 0.94, -0.15)	(2, 9)

Table 6.16: **Yeast exosome: statistics for U=0.4, F=0.6.**

					$\alpha = 0.25$		$\alpha = 0.50$		$\alpha = 0.75$		$\alpha = 1$	
oligomer size, s	$\lvert\text{Pool}_{\text{E}}(\mathcal{O}_{\leq s})\rvert$	$\lvert M\rvert$	mode	*sols type*	(ROC$_{sens.}$, ROC$_{spec.}$, Cvg, σ_{Cvg})	Solutions (#, size)	(ROC$_{sens.}$, ROC$_{spec.}$, Cvg, σ_{Cvg})	Solutions (#, size)	(ROC$_{sens.}$, ROC$_{spec.}$, Cvg, σ_{Cvg})	Solutions (#, size)	(ROC$_{sens.}$, ROC$_{spec.}$, Cvg, σ_{Cvg})	Solutions (#, size)
5	20	3	FU	all	(0.76, 1.00, 0.45, 0.18)	(3, 11)	(0.65, 1.00, 0.20, 0.13)	(8, 8)	(0.65, 1.00, 0.20, 0.13)	(8, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
5	20	3	FU	cons	(0.76, 1.00, 0.45, 0.18)	(3, 11)	(0.65, 1.00, 0.20, 0.13)	(8, 8)	(0.65, 1.00, 0.20, 0.13)	(8, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
5	20	3	FF	all	(0.71, 0.33, 0.20, 0.15)	(2, 13)	(0.59, 0.67, 0.10, 0.14)	(12, 8)	(0.59, 0.67, 0.10, 0.14)	(12, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
5	20	3	FF	cons	(0.71, 0.33, 0.20, 0.15)	(2, 13)	(0.59, 0.67, 0.10, 0.14)	(12, 8)	(0.59, 0.67, 0.10, 0.14)	(12, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
5	20	3	UF	all	(0.53, 0.33, -0.10, 0.15)	(6, 9)	(0.53, 0.33, -0.05, 0.14)	(6, 8)	(0.53, 0.33, -0.05, 0.14)	(6, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
5	20	3	UF	cons	(0.53, 0.33, -0.10, 0.15)	(6, 9)	(0.53, 0.33, -0.05, 0.14)	(6, 8)	(0.53, 0.33, -0.05, 0.14)	(6, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
5	20	3	UU	all	(0.59, 0.67, 0.10, 0.13)	(8, 8)	(0.59, 0.67, 0.10, 0.13)	(8, 8)	(0.59, 0.67, 0.10, 0.13)	(8, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
5	20	3	UU	cons	(0.59, 0.67, 0.10, 0.13)	(8, 8)	(0.59, 0.67, 0.10, 0.13)	(8, 8)	(0.59, 0.67, 0.10, 0.13)	(8, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
6	21	3	FU	all	(0.71, 1.00, 0.35, 0.19)	(2, 11)	(0.59, 1.00, 0.15, 0.13)	(4, 8)	(0.59, 1.00, 0.15, 0.13)	(4, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
6	21	3	FU	cons	(0.71, 1.00, 0.35, 0.19)	(2, 11)	(0.59, 1.00, 0.15, 0.13)	(4, 8)	(0.59, 1.00, 0.15, 0.13)	(4, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
6	21	3	FF	all	(0.71, 0.25, 0.20, 0.20)	(2, 13)	(0.59, 0.50, 0.05, 0.16)	(8, 8)	(0.59, 0.50, 0.05, 0.16)	(8, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
6	21	3	FF	cons	(0.71, 0.25, 0.20, 0.20)	(2, 13)	(0.59, 0.50, 0.05, 0.16)	(8, 8)	(0.59, 0.50, 0.05, 0.16)	(8, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
6	21	3	UF	all	(0.47, 0.25, -0.15, 0.17)	(6, 9)	(0.47, 0.50, -0.13, 0.15)	(6, 8)	(0.47, 0.50, -0.13, 0.15)	(6, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
6	21	3	UF	cons	(0.47, 0.25, -0.15, 0.17)	(6, 9)	(0.47, 0.50, -0.13, 0.15)	(6, 8)	(0.47, 0.50, -0.13, 0.15)	(6, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
6	21	3	UU	all	(0.59, 0.50, 0.13, 0.16)	(9, 8)	(0.59, 0.50, 0.13, 0.16)	(9, 8)	(0.59, 0.50, 0.13, 0.16)	(9, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
6	21	3	UU	cons	(0.59, 0.50, 0.13, 0.16)	(9, 8)	(0.59, 0.50, 0.13, 0.16)	(9, 8)	(0.59, 0.50, 0.13, 0.16)	(9, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
7	29	3	FU	all	(0.76, 1.00, 0.45, 0.15)	(2, 12)	(0.65, 1.00, 0.25, 0.11)	(6, 8)	(0.65, 1.00, 0.25, 0.11)	(6, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
7	29	3	FU	cons	(0.76, 1.00, 0.45, 0.15)	(2, 12)	(0.65, 1.00, 0.25, 0.11)	(6, 8)	(0.65, 1.00, 0.25, 0.11)	(6, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
7	29	3	FF	all	(0.73, 0.50, 0.08, 0.19)	(2, 17)	(0.59, 0.92, 0.15, 0.14)	(8, 8)	(0.59, 0.92, 0.15, 0.14)	(8, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
7	29	3	FF	cons	(0.73, 0.50, 0.08, 0.19)	(2, 17)	(0.59, 0.92, 0.15, 0.14)	(8, 8)	(0.59, 0.92, 0.15, 0.14)	(8, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
7	29	3	UF	all	(0.47, 0.42, -0.38, 0.16)	(6, 13)	(0.47, 0.83, -0.15, 0.14)	(6, 8)	(0.47, 0.83, -0.15, 0.14)	(6, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
7	29	3	UF	cons	(0.47, 0.42, -0.38, 0.16)	(6, 13)	(0.47, 0.83, -0.15, 0.14)	(6, 8)	(0.47, 0.83, -0.15, 0.14)	(6, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
7	29	3	UU	all	(0.62, 0.92, 0.15, 0.11)	(12, 8)	(0.62, 0.92, 0.15, 0.11)	(12, 8)	(0.62, 0.92, 0.15, 0.11)	(12, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
7	29	3	UU	cons	(0.62, 0.92, 0.15, 0.11)	(12, 8)	(0.62, 0.92, 0.15, 0.11)	(12, 8)	(0.62, 0.92, 0.15, 0.11)	(12, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
8	36	0	FU	all	(0.70, 1.00, 0.35, 0.18)	(4, 12)	(0.60, 1.00, 0.17, 0.19)	(6, 9)	(0.60, 1.00, 0.17, 0.19)	(6, 9)	(0.85, 0.69, 0.45, 0.00)	(274, 9)
8	36	0	FU	cons	(0.70, 1.00, 0.35, 0.18)	(4, 12)	(0.60, 1.00, 0.17, 0.19)	(6, 9)	(0.60, 1.00, 0.17, 0.19)	(6, 9)	(0.45, 0.94, -0.15, 0.00)	(2, 9)
8	36	0	FF	all	(0.70, 0.44, -0.05, 0.20)	(2, 21)	(0.55, 0.88, 0.00, 0.18)	(16, 9)	(0.55, 0.88, 0.00, 0.18)	(16, 9)	(0.85, 0.69, 0.45, 0.00)	(274, 9)
8	36	0	FF	cons	(0.70, 0.44, -0.05, 0.20)	(2, 21)	(0.55, 0.88, 0.00, 0.18)	(16, 9)	(0.55, 0.88, 0.00, 0.18)	(16, 9)	(0.45, 0.94, -0.15, 0.00)	(2, 9)
8	36	0	UF	all	(0.45, 0.38, -0.63, 0.16)	(6, 16)	(0.45, 0.81, -0.23, 0.15)	(7, 9)	(0.50, 0.81, -0.17, 0.13)	(8, 9)	(0.85, 0.69, 0.45, 0.00)	(274, 9)
8	36	0	UF	cons	(0.45, 0.38, -0.63, 0.16)	(6, 16)	(0.45, 0.81, -0.23, 0.15)	(7, 9)	(0.50, 0.81, -0.17, 0.13)	(8, 9)	(0.45, 0.94, -0.15, 0.00)	(2, 9)
8	36	0	UU	all	(0.55, 0.91, 0.05, 0.16)	(7, 9)	(0.55, 0.91, 0.05, 0.17)	(8, 9)	(0.55, 0.91, 0.05, 0.17)	(8, 9)	(0.85, 0.69, 0.45, 0.00)	(274, 9)
8	36	0	UU	cons	(0.55, 0.91, 0.05, 0.16)	(7, 9)	(0.55, 0.91, 0.05, 0.17)	(8, 9)	(0.55, 0.91, 0.05, 0.17)	(8, 9)	(0.45, 0.94, -0.15, 0.00)	(2, 9)

Table 6.17: **Yeast exosome: statistics for U=0.1, F=0.9, G=0.5.** 20 intances each.

					$\alpha = 0.25$		$\alpha = 0.50$		$\alpha = 0.75$		$\alpha = 1$	
oligomer size, s	$\lvert\text{Pool}_{\text{E}}(\mathcal{O}_{\leq s})\rvert$	$\lvert M\rvert$	mode	*sols type*	(ROC$_{sens.}$, ROC$_{spec.}$, Cvg, σ_{Cvg})	Solutions (#, size)	(ROC$_{sens.}$, ROC$_{spec.}$, Cvg, σ_{Cvg})	Solutions (#, size)	(ROC$_{sens.}$, ROC$_{spec.}$, Cvg, σ_{Cvg})	Solutions (#, size)	(ROC$_{sens.}$, ROC$_{spec.}$, Cvg, σ_{Cvg})	Solutions (#, size)
5	20	3	FU	all	(0.59, 1.00, 0.15, 0.12)	(3, 8)	(0.59, 1.00, 0.15, 0.12)	(3, 8)	(0.59, 1.00, 0.15, 0.12)	(3, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
5	20	3	FU	cons	(0.59, 1.00, 0.15, 0.12)	(3, 8)	(0.59, 1.00, 0.15, 0.12)	(3, 8)	(0.59, 1.00, 0.15, 0.12)	(3, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
5	20	3	FF	all	(0.59, 0.67, 0.10, 0.15)	(6, 8)	(0.59, 0.67, 0.10, 0.15)	(6, 8)	(0.59, 0.67, 0.10, 0.15)	(6, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
5	20	3	FF	cons	(0.59, 0.67, 0.10, 0.15)	(6, 8)	(0.59, 0.67, 0.10, 0.15)	(6, 8)	(0.59, 0.67, 0.10, 0.15)	(6, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
5	20	3	UF	all	(0.47, 0.33, -0.15, 0.12)	(4, 8)	(0.47, 0.33, -0.15, 0.12)	(4, 8)	(0.47, 0.33, -0.15, 0.12)	(4, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
5	20	3	UF	cons	(0.47, 0.33, -0.15, 0.12)	(4, 8)	(0.47, 0.33, -0.15, 0.12)	(4, 8)	(0.47, 0.33, -0.15, 0.12)	(4, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
5	20	3	UU	all	(0.56, 0.67, 0.05, 0.14)	(6, 8)	(0.56, 0.67, 0.05, 0.14)	(6, 8)	(0.56, 0.67, 0.05, 0.14)	(6, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
5	20	3	UU	cons	(0.56, 0.67, 0.05, 0.14)	(6, 8)	(0.56, 0.67, 0.05, 0.14)	(6, 8)	(0.56, 0.67, 0.05, 0.14)	(6, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
6	21	3	FU	all	(0.62, 1.00, 0.20, 0.13)	(6, 8)	(0.62, 1.00, 0.20, 0.13)	(6, 8)	(0.62, 1.00, 0.20, 0.13)	(6, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
6	21	3	FU	cons	(0.62, 1.00, 0.20, 0.13)	(6, 8)	(0.62, 1.00, 0.20, 0.13)	(6, 8)	(0.62, 1.00, 0.20, 0.13)	(6, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
6	21	3	FF	all	(0.59, 0.75, 0.05, 0.12)	(4, 8)	(0.59, 0.75, 0.05, 0.12)	(4, 8)	(0.59, 0.75, 0.05, 0.12)	(4, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
6	21	3	FF	cons	(0.59, 0.75, 0.05, 0.12)	(4, 8)	(0.59, 0.75, 0.05, 0.12)	(4, 8)	(0.59, 0.75, 0.05, 0.12)	(4, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
6	21	3	UF	all	(0.50, 0.50, -0.10, 0.16)	(7, 8)	(0.50, 0.50, -0.10, 0.16)	(7, 8)	(0.50, 0.50, -0.10, 0.16)	(7, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
6	21	3	UF	cons	(0.50, 0.50, -0.10, 0.16)	(7, 8)	(0.50, 0.50, -0.10, 0.16)	(7, 8)	(0.50, 0.50, -0.10, 0.16)	(7, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
6	21	3	UU	all	(0.59, 0.75, 0.10, 0.14)	(9, 8)	(0.59, 0.75, 0.10, 0.14)	(9, 8)	(0.59, 0.75, 0.10, 0.14)	(9, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
6	21	3	UU	cons	(0.59, 0.75, 0.10, 0.14)	(9, 8)	(0.59, 0.75, 0.10, 0.14)	(9, 8)	(0.59, 0.75, 0.10, 0.14)	(9, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
7	29	3	FU	all	(0.59, 1.00, 0.15, 0.11)	(5, 8)	(0.59, 1.00, 0.15, 0.11)	(5, 8)	(0.59, 1.00, 0.15, 0.11)	(5, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
7	29	3	FU	cons	(0.59, 1.00, 0.15, 0.11)	(5, 8)	(0.59, 1.00, 0.15, 0.11)	(5, 8)	(0.59, 1.00, 0.15, 0.11)	(5, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
7	29	3	FF	all	(0.59, 0.92, 0.13, 0.11)	(8, 8)	(0.59, 0.92, 0.13, 0.11)	(8, 8)	(0.59, 0.92, 0.13, 0.11)	(8, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
7	29	3	FF	cons	(0.59, 0.92, 0.13, 0.11)	(8, 8)	(0.59, 0.92, 0.13, 0.11)	(8, 8)	(0.59, 0.92, 0.13, 0.11)	(8, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
7	29	3	UF	all	(0.53, 0.83, -0.05, 0.16)	(6, 8)	(0.53, 0.83, -0.05, 0.16)	(6, 8)	(0.53, 0.83, -0.05, 0.16)	(6, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
7	29	3	UF	cons	(0.53, 0.83, -0.05, 0.16)	(6, 8)	(0.53, 0.83, -0.05, 0.16)	(6, 8)	(0.53, 0.83, -0.05, 0.16)	(6, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
7	29	3	UU	all	(0.59, 0.92, 0.10, 0.13)	(12, 8)	(0.59, 0.92, 0.10, 0.13)	(12, 8)	(0.59, 0.92, 0.10, 0.13)	(12, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
7	29	3	UU	cons	(0.59, 0.92, 0.10, 0.13)	(12, 8)	(0.59, 0.92, 0.10, 0.13)	(12, 8)	(0.59, 0.92, 0.10, 0.13)	(12, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
8	36	0	FU	all	(0.55, 1.00, 0.10, 0.18)	(4, 9)	(0.55, 1.00, 0.10, 0.18)	(4, 9)	(0.55, 1.00, 0.10, 0.18)	(4, 9)	(0.85, 0.69, 0.45, 0.00)	(274, 9)
8	36	0	FU	cons	(0.55, 1.00, 0.10, 0.18)	(4, 9)	(0.55, 1.00, 0.10, 0.18)	(4, 9)	(0.55, 1.00, 0.10, 0.18)	(4, 9)	(0.45, 0.94, -0.15, 0.00)	(2, 9)
8	36	0	FF	all	(0.55, 0.88, -0.05, 0.19)	(7, 9)	(0.55, 0.88, 0.03, 0.19)	(8, 9)	(0.55, 0.88, 0.03, 0.19)	(8, 9)	(0.85, 0.69, 0.45, 0.00)	(274, 9)
8	36	0	FF	cons	(0.55, 0.88, -0.05, 0.19)	(7, 9)	(0.55, 0.88, 0.03, 0.19)	(8, 9)	(0.55, 0.88, 0.03, 0.19)	(8, 9)	(0.45, 0.94, -0.15, 0.00)	(2, 9)
8	36	0	UF	all	(0.35, 0.75, -0.50, 0.18)	(4, 10)	(0.45, 0.81, -0.20, 0.17)	(4, 9)	(0.45, 0.81, -0.20, 0.17)	(4, 9)	(0.85, 0.69, 0.45, 0.00)	(274, 9)
8	36	0	UF	cons	(0.35, 0.75, -0.50, 0.18)	(4, 10)	(0.45, 0.81, -0.20, 0.17)	(4, 9)	(0.45, 0.81, -0.20, 0.17)	(4, 9)	(0.45, 0.94, -0.15, 0.00)	(2, 9)
8	36	0	UU	all	(0.55, 0.88, 0.08, 0.17)	(10, 9)	(0.55, 0.88, 0.08, 0.17)	(10, 9)	(0.55, 0.88, 0.08, 0.17)	(10, 9)	(0.85, 0.69, 0.45, 0.00)	(274, 9)
8	36	0	UU	cons	(0.55, 0.88, 0.08, 0.17)	(10, 9)	(0.55, 0.88, 0.08, 0.17)	(10, 9)	(0.55, 0.88, 0.08, 0.17)	(10, 9)	(0.45, 0.94, -0.15, 0.00)	(2, 9)

Table 6.18: **Yeast exosome: statistics for U=0.25, F=0.75, G=0.5.** 20 instances each.

					$\alpha = 0.25$		$\alpha = 0.50$		$\alpha = 0.75$		$\alpha = 1$	
oligomer size, s	$\lvert\text{Pool}_{\text{E}}(\mathcal{O}_{\leq s})\rvert$	$\lvert M\rvert$	mode	*sols type*	(ROC$_{sens.}$, ROC$_{spec.}$, Cvg, σ_{Cvg})	Solutions (#, size)	(ROC$_{sens.}$, ROC$_{spec.}$, Cvg, σ_{Cvg})	Solutions (#, size)	(ROC$_{sens.}$, ROC$_{spec.}$, Cvg, σ_{Cvg})	Solutions (#, size)	(ROC$_{sens.}$, ROC$_{spec.}$, Cvg, σ_{Cvg})	Solutions (#, size)
5	20	3	FU	all	(0.59, 1.00, 0.15, 0.12)	(8, 8)	(0.59, 1.00, 0.15, 0.12)	(8, 8)	(0.59, 1.00, 0.15, 0.12)	(8, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
5	20	3	FU	cons	(0.59, 1.00, 0.15, 0.12)	(8, 8)	(0.59, 1.00, 0.15, 0.12)	(8, 8)	(0.59, 1.00, 0.15, 0.12)	(8, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
5	20	3	FF	all	(0.59, 0.67, 0.10, 0.13)	(9, 8)	(0.59, 0.67, 0.10, 0.13)	(9, 8)	(0.59, 0.67, 0.10, 0.13)	(9, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
5	20	3	FF	cons	(0.59, 0.67, 0.10, 0.13)	(9, 8)	(0.59, 0.67, 0.10, 0.13)	(9, 8)	(0.59, 0.67, 0.10, 0.13)	(9, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
5	20	3	UF	all	(0.47, 0.33, -0.15, 0.14)	(4, 8)	(0.47, 0.33, -0.15, 0.14)	(4, 8)	(0.47, 0.33, -0.15, 0.14)	(4, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
5	20	3	UF	cons	(0.47, 0.33, -0.15, 0.14)	(4, 8)	(0.47, 0.33, -0.15, 0.14)	(4, 8)	(0.47, 0.33, -0.15, 0.14)	(4, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
5	20	3	UU	all	(0.59, 0.67, 0.05, 0.12)	(6, 8)	(0.59, 0.67, 0.05, 0.12)	(6, 8)	(0.59, 0.67, 0.05, 0.12)	(6, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
5	20	3	UU	cons	(0.59, 0.67, 0.05, 0.12)	(6, 8)	(0.59, 0.67, 0.05, 0.12)	(6, 8)	(0.59, 0.67, 0.05, 0.12)	(6, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
6	21	3	FU	all	(0.59, 1.00, 0.15, 0.13)	(6, 8)	(0.59, 1.00, 0.15, 0.13)	(6, 8)	(0.59, 1.00, 0.15, 0.13)	(6, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
6	21	3	FU	cons	(0.59, 1.00, 0.15, 0.13)	(6, 8)	(0.59, 1.00, 0.15, 0.13)	(6, 8)	(0.59, 1.00, 0.15, 0.13)	(6, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
6	21	3	FF	all	(0.59, 0.75, 0.10, 0.15)	(11, 8)	(0.59, 0.75, 0.10, 0.15)	(11, 8)	(0.59, 0.75, 0.10, 0.15)	(11, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
6	21	3	FF	cons	(0.59, 0.75, 0.10, 0.15)	(11, 8)	(0.59, 0.75, 0.10, 0.15)	(11, 8)	(0.59, 0.75, 0.10, 0.15)	(11, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
6	21	3	UF	all	(0.47, 0.50, -0.15, 0.13)	(4, 8)	(0.47, 0.50, -0.15, 0.13)	(4, 8)	(0.47, 0.50, -0.15, 0.13)	(4, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
6	21	3	UF	cons	(0.47, 0.50, -0.15, 0.13)	(4, 8)	(0.47, 0.50, -0.15, 0.13)	(4, 8)	(0.47, 0.50, -0.15, 0.13)	(4, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
6	21	3	UU	all	(0.59, 0.75, 0.13, 0.14)	(9, 8)	(0.59, 0.75, 0.13, 0.14)	(9, 8)	(0.59, 0.75, 0.13, 0.14)	(9, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
6	21	3	UU	cons	(0.59, 0.75, 0.13, 0.14)	(9, 8)	(0.59, 0.75, 0.13, 0.14)	(9, 8)	(0.59, 0.75, 0.13, 0.14)	(9, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
7	29	3	FU	all	(0.59, 1.00, 0.15, 0.12)	(6, 8)	(0.59, 1.00, 0.15, 0.12)	(6, 8)	(0.59, 1.00, 0.15, 0.12)	(6, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
7	29	3	FU	cons	(0.59, 1.00, 0.15, 0.12)	(6, 8)	(0.59, 1.00, 0.15, 0.12)	(6, 8)	(0.59, 1.00, 0.15, 0.12)	(6, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
7	29	3	FF	all	(0.59, 0.92, 0.10, 0.12)	(7, 8)	(0.59, 0.92, 0.10, 0.12)	(7, 8)	(0.59, 0.92, 0.10, 0.12)	(7, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
7	29	3	FF	cons	(0.59, 0.92, 0.10, 0.12)	(7, 8)	(0.59, 0.92, 0.10, 0.12)	(7, 8)	(0.59, 0.92, 0.10, 0.12)	(7, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
7	29	3	UF	all	(0.53, 0.83, -0.05, 0.15)	(6, 8)	(0.53, 0.83, -0.05, 0.15)	(6, 8)	(0.53, 0.83, -0.05, 0.15)	(6, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
7	29	3	UF	cons	(0.53, 0.83, -0.05, 0.15)	(6, 8)	(0.53, 0.83, -0.05, 0.15)	(6, 8)	(0.53, 0.83, -0.05, 0.15)	(6, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
7	29	3	UU	all	(0.53, 0.92, 0.05, 0.14)	(7, 8)	(0.53, 0.92, 0.05, 0.14)	(7, 8)	(0.53, 0.92, 0.05, 0.14)	(7, 8)	(0.76, 0.33, 0.35, 0.00)	(96, 8)
7	29	3	UU	cons	(0.53, 0.92, 0.05, 0.14)	(7, 8)	(0.53, 0.92, 0.05, 0.14)	(7, 8)	(0.53, 0.92, 0.05, 0.14)	(7, 8)	(0.76, 0.33, 0.35, 0.00)	(48, 8)
8	36	0	FU	all	(0.55, 1.00, 0.10, 0.19)	(4, 9)	(0.55, 1.00, 0.10, 0.19)	(4, 9)	(0.55, 1.00, 0.10, 0.19)	(4, 9)	(0.85, 0.69, 0.45, 0.00)	(274, 9)
8	36	0	FU	cons	(0.55, 1.00, 0.10, 0.19)	(4, 9)	(0.55, 1.00, 0.10, 0.19)	(4, 9)	(0.55, 1.00, 0.10, 0.19)	(4, 9)	(0.45, 0.94, -0.15, 0.00)	(2, 9)
8	36	0	FF	all	(0.55, 0.88, -0.05, 0.17)	(8, 9)	(0.55, 0.88, -0.05, 0.17)	(8, 9)	(0.55, 0.88, -0.05, 0.17)	(8, 9)	(0.85, 0.69, 0.45, 0.00)	(274, 9)
8	36	0	FF	cons	(0.55, 0.88, -0.05, 0.17)	(8, 9)	(0.55, 0.88, -0.05, 0.17)	(8, 9)	(0.55, 0.88, -0.05, 0.17)	(8, 9)	(0.45, 0.94, -0.15, 0.00)	(2, 9)
8	36	0	UF	all	(0.50, 0.81, -0.15, 0.14)	(9, 9)	(0.50, 0.81, -0.15, 0.14)	(9, 9)	(0.50, 0.81, -0.15, 0.14)	(9, 9)	(0.85, 0.69, 0.45, 0.00)	(274, 9)
8	36	0	UF	cons	(0.50, 0.81, -0.15, 0.14)	(9, 9)	(0.50, 0.81, -0.15, 0.14)	(9, 9)	(0.50, 0.81, -0.15, 0.14)	(9, 9)	(0.45, 0.94, -0.15, 0.00)	(2, 9)
8	36	0	UU	all	(0.60, 0.88, 0.08, 0.13)	(12, 9)	(0.60, 0.88, 0.08, 0.13)	(12, 9)	(0.60, 0.88, 0.08, 0.13)	(12, 9)	(0.85, 0.69, 0.45, 0.00)	(274, 9)
8	36	0	UU	cons	(0.60, 0.88, 0.08, 0.13)	(12, 9)	(0.60, 0.88, 0.08, 0.13)	(12, 9)	(0.60, 0.88, 0.08, 0.13)	(12, 9)	(0.45, 0.94, -0.15, 0.00)	(2, 9)

Table 6.19: **Yeast exosome: statistics for U=0.4, F=0.6, G=0.5.** 20 instances each.

Chapter 7

Conclusion

Mass spectrometry has emerged as a powerful technique in the last decade to provide crucial low resolution information particularly on large macromolecular assemblies in the absence of elusive high resolution information provided by X-ray crystallography. In this thesis, we have focused in detail, in particular, on two problems in pursuit of structural inference of large macromolecular assemblies. The first one, *stoichiometry determination*, consists of computing the number of copies of each protein the assembly is composed of, taking into account the error in mass measurement. The second one, *connectivity inference*, aims at inferring pairwise interactions among its proteins responsible to hold the assembly intact. The algorithms we provide are efficient both in terms of space and time and show large improvement in performance over previous work.

The problem of stoichiometry determination is the first problem to be resolved given the mass measurements of each composing proteins and assembly as a whole, thanks to the native mass spectrometry. We have developed a constant memory space enumeration algorithm (`DIOPHANTINE`), and an output sensitive dynamic programming based algorithm (`DP++`), both for the interval case, which outperforms state-of-the-art stoichiometry determination algorithms by several (three to four) orders of magnitude when run on MS data for large macromolecular assemblies. We observe that for nominal error value in the target mass, the problem yields large number of solution rendering the enumeration of the solutions relevant. We also test our algorithm on metabolite datasets and find the performance satisfactory as compared to the previous work. In order to take into account the fractional values of weights, it was previously suggested to redefine the problem utilizing a blow up factor, b. Also, one had to determine an optimum value of b to avoid doing calculations giving false positives which had to be otherwise weed out. Our algorithm DIOPHANTINE requires only to multiply all the weights by an appropriate power of 10 (depending upon the precision of measurement). The problem then resembles the standard money changing problem. Thus, this simple algorithm DIOPHANTINE is constant memory and spares the user of any complicated pre-treatment of the input.

Once the composition of assembly is determined one would like to report the pairwise interactions among proteins holding the topology of the assembly intact. In this regard, we solve the problem on connectivity inference, given the population of oligomers (sub-complexes) of different sizes generated by controlled dissociation of the assembly. On the theoretical side, the complexity of the problem is discussed. We propose two algorithms, a greedy strategy and a mixed integer programming algorithm. On the application side, we solve *Minimum Connectivity Inference* problem in which we generate all optimal solutions accounting for a given set of oligomers. Here optimality is pertaining to the most parsimonious set of contacts solving for the population of oligomers generated out of an assembly. We observe that each optimal solution is marked with high precision for the test case of yeast exosome, for which the exhaustive set of contacts seen in the crystal structure are at disposal. We also define a subset of these solutions called consensus solutions, which are composed of the most frequent contacts in the union of all solutions. These consensus contacts are the

backbone of the complete solution set and are observed to have very high precision, i.e. almost no false positives. The algorithm also provides option to compute all possible solutions have an user defined solution size.

The above connectivity inference problem is solved with an assumption that no two contacts have an additional preference over the other in case there is a tie at some stage of the algorithm, hence all possible solutions are reported. In practical scenarios, one may have apriori information on some of the contacts that is obtained from other experiments such as cross linking experiments, yeast two hybrid essays, co-immunoprecipitation essays etc. These experiments could provide information on a pairwise interaction being more probable over the other.

In order to take into account the relative biases of some contacts , we introduce the *Minimum Weight Connectivity Inference* problem (MWCI), which generalize the *Minimum Connectivity Inference* problem, by introducing weights associated with putative contacts. The weight of a putative contact actually has two components: a *fixed* one which is identical for all contacts, and a *variable* one, set on a per contact basis. For the variable weight, a default baseline weight is chosen to be 0.5 and higher value in the range (0.5, 1) is assigned to the more probable contacts. A value in the range (0, 0.5) could be assigned to the contacts which are relatively less probable. For instance, if one has an electron microscopy map of an assembly, those pair of proteins eclipsed by other proteins or are far apart are less likely to interact, hence they can either be assigned a low value between (0,0.5) or could even be labeled *Forbidden*, 'F', to rule out their sampling altogether. The balance between fixed and variable weights is governed by a parameter α between 0 and 1. For $\alpha = 1$, only the constant weights matter, so that a MWCI problem with $\alpha = 1$ reduces to a MCI problem. Again, the individual weights and the value of α used allow the user to incorporate the a priori information available on the pairwise interactions and the contacts.

We also develop an algorithm, `MILP-W` to solve MWCI problems, and explore the incidence of the parameter α and of variable weights. The effect of weights is visible at lower value of α. As expected, the contacts with higher assigned weights populate the solutions. Those solutions with contacts having low assigned weights get low priority and are thus ruled out. In an another experiment, with the test example of yeast exosome, we observe that if weights of about half of the contacts chosen randomly in fair coin tossing experiment are set to a higher value then the sensitivity is high, whereas, if the weights are set to a lower value then the specificity is observed to be high.

Finally, we develop an algorithm, MILP-$\mathrm{W_B}$, a bootstrap strategy aiming at enriching the solutions reported by MILP-W. We start with a set of contacts in the union of consensus solutions, since the precision is excellent. The idea is to obtain consequent set of consensus contacts on barring the initial set of consensus contacts ($\mathcal{E}^{cons.}_{\mathrm{MILP}}$) from the pool of contacts. Algorithmically, these contacts are labeled as *Forbidden*, F (or given lower weight, e.g., 0.1). This triggers alternative connectivity for the same set of oligomers. The process is begun by forbidding one contact at a time from initial consensus contacts (except the dimers) and one reports the union of consensus contacts yielded after all the `MILP-W` (`MILP`, for $\alpha = 1$) runs corresponding to each contacts being forbidden. The process can be iterated by precluding two or more contacts at a time but at possible expense of induction of more false-positives. We examine the performance of the algorithm `MILP-`$\mathrm{W_B}$ on three test systems – yeast exosome, yeast proteasome lid and human eIF3. It is observed that the sensitivity yielded is almost twice of that published previously by *Network Inference* algorithm by Robinson et al. The coverage scores are improved significantly as well. Also, pushing the bootstrapping procedure beyond preclusion of 1 contact at a time increases sensitivity at the expense of drop in specificity. The coverage score, subsequently, rises initially and then begins to fall.

We believe that our algorithms would prove to be useful to the structural biology community at this juncture when, due to the limitation of high resolution acquisition other biophysical experimental techniques are utilized for large macromolecular assemblies to unveil its different aspects.

We make the software freely available in the interest of knowledge sharing and for valuable suggestions for the improvement of this work.

Bibliography

[AAC+13] D. Agarwal, J. Araujo, C. Caillouet, F. Cazals, D. Coudert, and S. Pérennes. Connectivity inference in mass spectrometry based structure determination. In H.L. Bodlaender and G.F. Italiano, editors, *European Symposium on Algorithms (Springer LNCS 8125)*, pages 289–300, Sophia Antipolis, France, 2013. Springer.

[ACL+02] P. Aloy, F. D. Ciccarelli, C. Leutwein, A. C. Gavin, G. Superti-Furga, P. Bork, B. Böttcher, and R. B. Russell. A complex prediction: three-dimensional model of the yeast exosome. *EMBO reports*, 3(7):628–635, 2002.

[ACMD14] D. Agarwal, F. Cazals, and N. Malod-Dognin. Stoichiometry determination for mass-spectrometry data: the interval case. *Submitted*, 2014. Preprint: Inria tech report 8101.

[ADV+07] F. Alber, S. Dokudovskaya, L. M. Veenhoff, W. Zhang, J. Kipper, D. Devos, A. Suprapto, O. Karni-Schmidt, R. Williams, B.T. Chait, M.P. Rout, and A. Sali. Determining the Architectures of Macromolecular Assemblies. *Nature*, 450(7170):683–694, Nov 2007.

[AFK+08] F. Alber, F. Förster, D. Korkin, M. Topf, and A. Sali. Integrating diverse data for structure determination of macromolecular assemblies. *Ann. Rev. Biochem.*, 77:11.1–11.35, 2008.

[Agn02] G. Agnarsson. On the sylvester denumerants for general restricted partitions. *Congressus numerantium*, pages 49–60, 2002.

[Alf05] J.L. Ramírez Alfonsín. *The Diophantine Frobenius Problem.* Oxford University Press, 2005.

[AMS06] Noga Alon, Dana Moshkovitz, and Shmuel Safra. Algorithmic construction of sets for k-restrictions. *ACM Trans. Algorithms*, 2:153–177, April 2006.

[BBB+10] Richard Baran, Benjamin P Bowen, Nicholas J Bouskill, Eoin L Brodie, Steven M Yannone, and Trent R Northen. Metabolite identification in synechococcus sp. pcc 7002 using untargeted stable isotope assisted metabolite profiling. *Analytical chemistry*, 82(21):9034–9042, 2010.

[Bd72] A.G. Beged-dov. Lower and upper bounds for the number of lattice points in a simplex. *SIAM Journal on Applied Mathematics*, 22(1):106–108, 1972.

[BDBW11] Anne E Blackwell, Eric D Dodds, Vahe Bandarian, and Vicki H Wysocki. Revealing the quaternary structure of a heterogeneous noncovalent protein complex through surface-induced dissociation. *Analytical chemistry*, 83(8):2862–2865, 2011.

[BDF+10] A. Brown, E. Dannenberg, J. Fox, J. Hanna, K. Keck, A. Moore, Z. Robbins, B. Samples, and J. Stankewicz. On a generalization of the frobenius number. *Journal of Integer Sequences*, 13(2):3, 2010.

[Bel43] E.T. Bell. Interpolated denumerants and lambert series. *American Journal of Mathematics*, 65(3):382–386, 1943.

[Bel57] R.E. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[BL05a] S. Bocker and Z. Liptak. Efficient mass decomposition. In ACM, editor, *ACM Symposium on Applied Computing*, 2005.

[BL05b] S. Böcker and Z. Lipták. The money changing problem revisited: Computing the frobenius number in time $o(ka1)^*$. In Lusheng Wang, editor, *Computing and Combinatorics*, volume 3595 of *Lecture Notes in Computer Science*, pages 965–974. Springer Berlin / Heidelberg, 2005.

[BL07] S. Bocker and Z. Liptak. A fast and simple algorithm for the money changing problem. *Algorithmica*, 48(4):413–432, 2007.

[BLJ07] Tord Berggård, Sara Linse, and Peter James. Methods for the detection and analysis of protein–protein interactions. *Proteomics*, 7(16):2833–2842, 2007.

[BLLP06] S. Böcker, M. Letzel, Z. Lipták, and A. Pervukhin. Decomposing metabolomic isotope patterns. *Algorithms in Bioinformatics*, 4175:12–23, 2006.

[BLLP09] Sebastian Böcker, Matthias C Letzel, Zsuzsanna Lipták, and Anton Pervukhin. Sirius: decomposing isotope patterns for metabolite identification. *Bioinformatics*, 25(2):218–224, 2009.

[BLM+08] S. Böcker, Z. Lipták, M. Martin, A. Pervukhin, and H. Sudek. Decomp – from interpreting mass spectrometry peaks to solving the money changing problem. *Bioinformatics*, 24(4):591–593, 2008.

[BM08] A. Bondy and U. S. R. Murty. *Graph Theory*, volume 244 of *Graduate Texts in Mathematics*. Springer, 2008.

[BM12] Shibdas Banerjee and Shyamalava Mazumdar. Electrospray ionization mass spectrometry: a technique to access the information beyond the molecular weight of the analyte. *International journal of analytical chemistry*, 2012, 2012.

[BR11] J.L.P. Benesch and B.T. Ruotolo. Mass spectrometry: come of age for structural and dynamical biology. *Current opinion in structural biology*, 21:641–649, 2011.

[BSG59] K Biemann, J Seibl, and F Gapp. Mass spectrometric identification of amino acids. *Biochemical and Biophysical Research Communications*, 1(6):307–311, 1959.

[BTD+08] P. Braun, M. Tasan, M. Dreze, M. Barrios-Rodiles, I. Lemmens, H. Yu, J. Sahalie, R. Murray, L. Roncari, A-S. De Smet, K. Venketesan, J-F. Rual, J. Vandenhaute, M.E. Cusick, T. Pawson, D.E. Hill, J. Tavernier, J.L. Wrana, F.P. Roth, and M. Vidal. An experimentally derived confidence score for binary protein-protein interactions. *Nature methods*, 6(1):91–97, 2008.

[CAB+04] Steven Carr, Ruedi Aebersold, Michael Baldwin, AL Burlingame, Karl Clauser, and Alexey Nesvizhskii. The need for guidelines in publication of peptide and protein identification data working group on publication guidelines for peptide and protein identification data. *Molecular & Cellular Proteomics*, 3(6):531–533, 2004.

[CGH+13] Fan Chen, Sabina Gerber, Katrin Heuser, Vladimir M Korkhov, Christian Lizak, Samantha Mireku, Kaspar P Locher, and Renato Zenobi. High-mass matrix-assisted laser desorption ionization-mass spectrometry of integral membrane proteins and their complexes. *Analytical chemistry*, 85(7):3483–3488, 2013.

[CK08] F. Cazals and C. Karande. A note on the problem of reporting maximal cliques. *Theoretical Computer Science*, 407(1–3):564–568, 2008.

[Com74] L. Comtet. *Advanced Combinatorics: The art of finite and infinite expansions*. Not Avail, 1974.

[CPBJ06] F. Cazals, F. Proust, R. Bahadur, and J. Janin. Revisiting the Voronoi description of protein-protein interfaces. *Protein Science*, 15(9):2082–2092, 2006.

[DAH07] Katja Dettmer, Pavel A Aronov, and Bruce D Hammock. Mass spectrometry-based metabolomics. *Mass spectrometry reviews*, 26(1):51–78, 2007.

[DDC12] T. Dreyfus, V. Doye, and F. Cazals. Assessing the reconstruction of macro-molecular assemblies with toleranced models. *Proteins: structure, function, and bioinformatics*, 80(9):2125–2136, 2012.

[DDC13] T. Dreyfus, V. Doye, and F. Cazals. Probing a continuum of macro-molecular assembly models with graph templates of sub-complexes. *Proteins: structure, function, and bioinformatics*, 81(11):2034–2044, 2013.

[DF80] R. G. Dromey and G. T. Foyster. Calculation of elemental compositions from high resolution mass spectral data. *Analytical Chemistry*, 52(3):394–398, 1980.

[DFZ+07] E. Damoc, C. S. Fraser, M. Zhou, H. Videler, G. L. Mayeur, J. W. B. Hershey, J. A. Doudna, C. V. Robinson, and J. A. Leary. Structural characterization of the human eukaryotic initiation factor 3 protein complex by mass spectrometry. *Molecular & Cellular Proteomics*, 6(7):1135–1146, 2007.

[DH08] M.A. D'Angelo and M.W. Hetzer. Structure, dynamics and function of nuclear pore complexes. *Trends Cell Biology*, 18:456–522, 2008.

[DLMB13] K. Dührkop, M. Ludwig, M. Meusel, and S. Böcker. Faster mass decomposition. In *WABI*, Sophia-Antipolis, 2013.

[Dyd10] Fred Dyda. Developments in low-resolution biological x-ray crystallography. *F1000 biology reports*, 2, 2010.

[ea01] O. Puig et al. The tandem affinity purification method: A general procedure of protein complex purification. *Methods*, 24:218–229, 2001.

[ES07] Hoffmann Ed and Vincent Stroobant. Mass spectrometry: principles and applications. *England: West Sussex*, 2007.

[Fer99] A. Fersht. *Structure and Mechanism in Protein Science: A Guide to Enzyme Catalysis and Protein Folding*. Freeman, 1999.

[FMM+89] John B Fenn, Matthias Mann, Chin Kai Meng, Shek Fu Wong, and Craig M Whitehouse. Electrospray ionization for mass spectrometry of large biomolecules. *Science*, 246(4926):64–71, 1989.

[FMPS+12] J. Fernandez-Martinez, J. Phillips, M.D. Sekedat, R. Diaz-Avalos, J. Velazquez-Muriel, J.D. Franke, R. Williams, D.L. Stokes, B.T. Chait, A. Sali, and M.P. Rout. Structure–function mapping of a heptameric module in the nuclear pore complex. *The Journal of Cell Biology*, 2012.

[FS09] P. Flajolet and R. Sedgewick. *Analytic combinatorics*. Cambridge Univ Pr, 2009.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

[GKM+11] Parikshit Gopalan, Adam Klivans, Raghu Meka, Daniel Stefankovic, Santosh Vempala, and Eric Vigoda. An fptas for# knapsack and related counting problems. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 817–826. IEEE, 2011.

[GLM+11] Patrick Giavalisco, Yan Li, Annemarie Matthes, Aenne Eckhardt, Hans-Michael Hubberten, Holger Hesse, Shruthi Segu, Jan Hummel, Karin Köhl, and Lothar Willmitzer. Elemental formula annotation of polar and lipophilic metabolites using 13c, 15n and 34s isotope labelling, in combination with high-resolution mass spectrometry. *The Plant Journal*, 68(2):364–376, 2011.

[GRDP09] Valérie Gabelica, Frédéric Rosu, and Edwin De Pauw. A simple method to determine electrospray response factors of noncovalent complexes. *Analytical chemistry*, 81(16):6708–6715, 2009.

[HAK+10] Hisayuki Horai, Masanori Arita, Shigehiko Kanaya, Yoshito Nihei, Tasuku Ikeda, Kazuhiro Suwa, Yuya Ojima, Kenichi Tanaka, Satoshi Tanaka, Ken Aoshima, et al. Massbank: a public repository for sharing mass spectral data for life sciences. *Journal of mass spectrometry*, 45(7):703–714, 2010.

[HDT+06] H. Hernández, A. Dziembowski, T. Taverner, B. Séraphin, and C. V. Robinson. Subunit architecture of multimeric complexes isolated directly from cells. *EMBO reports*, 7(6):605–610, 2006.

[HSC+07] Adrian D Hegeman, Christopher F Schulte, Qiu Cui, Ian A Lewis, Edward L Huttlin, Hamid Eghbalnia, Amy C Harms, Eldon L Ulrich, John L Markley, and Michael R Sussman. Stable isotope assisted assignment of elemental compositions for metabolomics. *Analytical chemistry*, 79(18):6912–6921, 2007.

[JBC08] J. Janin, R. P. Bahadur, and P. Chakrabarti. Protein-protein interaction and quaternary structure. *Quarterly reviews of biophysics*, 41(2):133–180, 2008.

[JHP10] R. J. Jackson, C. U. T. Hellen, and T. V. Pestova. The mechanism of eukaryotic translation initiation and principles of its regulation. *Nature Reviews Molecular Cell Biology*, 11(2):113–127, 2010.

[Kar72] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, March 1972.

[Kel04] N.L. Kelleher. Peer reviewed: Top-down proteomics. *Analytical chemistry*, 76(11):196–203, 2004.

[KF06] Tobias Kind and Oliver Fiehn. Metabolomic database annotations via query of elemental compositions: mass accuracy is insufficient even at less than 1 ppm. *BMC bioinformatics*, 7(1):234, 2006.

[Kin23] KH Kingdon. A method for the neutralization of electron space charge by positive ionization at very low gas pressures. *Physical Review*, 21(4):408, 1923.

[Kni81] RD Knight. Storage of ions from laser-produced plasmas. *Applied Physics Letters*, 38(4):221–223, 1981.

[KRCM+12] Piotr T Kasper, Miguel Rojas-Chertó, Robert Mistrik, Theo Reijmers, Thomas Hankemeier, and Rob J Vreeken. Fragmentation trees for the structural characterisation of metabolites. *Rapid Communications in Mass Spectrometry*, 26(19):2275–2286, 2012.

[KRY+12] Athit Kao, Arlo Randall, Yingying Yang, Vishal R Patel, Wynne Kandur, Shenheng Guan, Scott D Rychnovsky, Pierre Baldi, and Lan Huang. Mapping the structural topology of the yeast 19s proteasomal regulatory particle using chemical cross-linking and probabilistic modeling. *Molecular & Cellular Proteomics*, 11:1566–1577, 2012.

[Lam74] T.A. Lambe. Bounds on the number of feasible solutions to a knapsack problem. *SIAM Journal on Applied Mathematics*, 26(2):302–305, 1974.

[LBC+08] Steven J Ludtke, Matthew L Baker, Dong-Hua Chen, Jiu-Li Song, David T Chuang, and Wah Chiu. De novo backbone trace of groel from single particle electron cryomicroscopy. *Structure*, 16(3):441–448, 2008.

[LBERT08] E. Levy, E. Boeri-Erba, C. Robinson, and S. Teichmann. Assembly reflects evolution of protein complexes. *Nature*, 453(7199):1262–1265, 2008.

[LC10] S. Loriot and F. Cazals. Modeling macro-molecular interfaces with Intervor. *Bioinformatics*, 26(7):964–965, 2010.

[LEM+12] Gabriel C Lander, Eric Estrin, Mary E Matyskiela, Charlene Bashore, Eva Nogales, and Andreas Martin. Complete subunit architecture of the proteasome regulatory particle. *Nature*, 482(7384):186–191, 2012.

[LFB+12] K. Lasker, F. Förster, S. Bohn, T. Walzthoeni, E. Villa, P. Unverdorben, F. Beck, R. Aebersold, A. Sali, and W. Baumeister. Molecular architecture of the 26s proteasome holocomplex determined by an integrative approach. *PNAS*, 109(5):1380–1387, 2012.

[Loo97] Joseph A Loo. Studying noncovalent protein complexes by electrospray ionization mass spectrometry. *Mass Spectrometry Reviews*, 16(1):1–23, 1997.

[LSS+94] Daniel P Little, J Paul Speir, Michael W Senko, Peter B O'Connor, and Fred W McLafferty. Infrared multiphoton dissociation of large multiply charged ions for biomolecule sequencing. *Analytical Chemistry*, 66(18):2809–2815, 1994.

[LY94] Carsten Lund and Mihalis Yannakakis. On the hardness of approximating minimization problems. *J. ACM*, 41:960–981, September 1994.

[Mak00] Alexander Makarov. Electrostatic axially harmonic orbital trapping: a high-performance technique of mass analysis. *Analytical chemistry*, 72(6):1156–1162, 2000.

[MBC13] Debora Lika Makino, Marc Baumgärtner, and Elena Conti. Crystal structure of an rna-bound 11-subunit eukaryotic exosome complex. *Nature*, 495(7439):70–75, 2013.

[MHFH10] R. Mahmoudvand, H. Hassani, A. Farzaneh, and G. Howell. The exact number of nonnegative integer solutions for a linear diophantine inequality. *IAENG International Journal of Applied Mathematics*, 40(1), 2010.

[MM88] Edward A Mason and Earl W McDaniel. Transport properties of ions in gases. *NASA STI/Recon Technical Report A*, 89:15174, 1988.

[MM15] Jody C May and John A McLean. Ion mobility-mass spectrometry: time-dispersive instrumentation. *Analytical chemistry*, 87(3):1422–1436, 2015.

[MR12] N. Morgner and C. V. Robinson. Massign: An assignment strategy for maximising information from the mass spectra of heterogeneous protein assemblies. *Analytical Chemistry*, 84(6):2939–2948, 2012.

[MYL02] Roger E Moore, Mary K Young, and Terry D Lee. Qscore: an algorithm for evaluating sequest database search results. *Journal of the American Society for Mass Spectrometry*, 13(4):378–386, 2002.

[NB10] Steffen Neumann and Sebastian Böcker. Computational mass spectrometry for metabolomics – a review. *Anal Bioanal Chem*, 398(7):2779–2788, 2010.

[Pad71] M.W. Padberg. A remark on "an inequality for the number of lattice points in a simplex". *SIAM Journal on Applied Mathematics*, 20(4):638–641, 1971.

[PCN08] Richard H Perry, R Graham Cooks, and Robert J Noll. Orbitrap mass spectrometry: instrumentation, ion motion and applications. *Mass spectrometry reviews*, 27(6):661–699, 2008.

[PCR+01] Oscar Puig, Friederike Caspary, Guillaume Rigaut, Berthold Rutz, Emmanuelle Bouveret, Elisabeth Bragado-Nilsson, Matthias Wilm, and Bertrand Séraphin. The tandem affinity purification (tap) method: a general procedure of protein complex purification. *Methods*, 24(3):218–229, 2001.

[Pis05] D. Pisinger. Where are the hard knapsack problems? *Computers and Operations research*, 32:2271–2284, 2005.

[PKP04] U. Pferschy, H. Kellerer, and D. Pisinger. *Knapsack Problems*. Springer, 2004.

[PYA09] V. Poirriez, N. Yanev, and R. Andonov. A hybrid algorithm for the unbounded knapsack problem. *Discrete Optimization*, 6(1):110 – 124, 2009.

[QA10] Ermir Qeli and Christian H Ahrens. Peptideclassifier for protein inference and targeted quantitative proteomics. *Nature biotechnology*, 28(7):647–650, 2010.

[QASV+13] Jordi Querol-Audi, Chaomin Sun, Jacob M Vogan, M Duane Smith, Yu Gu, Jamie HD Cate, and Eva Nogales. Architecture of human translation initiation factor 3. *Structure*, 21(6):920–928, 2013.

[RA96] J. L. Ramírez-Alfonsín. Complexity of the frobenius problem. *Combinatorica*, 16:143–147, 1996. 10.1007/BF01300131.

[Rag95] S. Raghavan. *Formulations and algorithms for network design problems with connectivity requirements*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1995.

[RBHM00] Ryan P Rodgers, Erin N Blumer, Christopher L Hendrickson, and Alan G Marshall. Stable isotope incorporation triples the upper mass limit for determination of elemental composition by accurate mass measurement. *Journal of the American Society for Mass Spectrometry*, 11(10):835–840, 2000.

[RDD+12] R. J. Rose, E. Damoc, E. Denisov, A. Makarov, and A. J. R. Heck. High-sensitivity orbitrap mass analysis of intact macromolecular assemblies. *Nature Methods*, 9(11):1084–1086, 2012.

[RSB07] Carol V Robinson, Andrej Sali, and Wolfgang Baumeister. The molecular sociology of the cell. *Nature*, 450(7172):973–982, 2007.

[SAR12] Florian Stengel, Ruedi Aebersold, and Carol V Robinson. Joining forces: integrating proteomics and cross-linking with the mass spectrometry of intact complexes. *Molecular & Cellular Proteomics*, 11(3), 2012.

[SH14] Joost Snijder and Albert JR Heck. Analytical approaches for size and mass analysis of large protein assemblies. *Annual Review of Analytical Chemistry*, 7:43–64, 2014.

[Sha10] Michal Sharon. How far can we go with structural mass spectrometry of protein complexes? *Journal of the American Society for Mass Spectrometry*, 21(4):487–500, 2010.

[Sma98] N.P. Smart. *The algorithmic resolution of Diophantine equations*, volume 41. Cambridge Univ Pr, 1998.

[SMBE+09] M. Sharon, H. Mao, E. Boeri Erba, E. Stephens, N. Zheng, and C.V. Robinson. Symmetrical modularity of the cop9 signalosome complex suggests its multifunctionality. *Structure*, 17(1):31–40, 2009.

[SR07] M. Sharon and C.V. Robinson. The role of mass spectrometry in structure elucidation of dynamic protein complexes. *Annu. Rev. Biochem.*, 76:167–193, 2007.

[SR11] Michal Sharon and Carol V Robinson. Peeling back the layers of complexity. *Current opinion in structural biology*, 21(5):619–621, 2011.

[SS11] J. Shallit and J. Stankewicz. Unbounded discrepancy in frobenius numbers. *Integers*, 11:1–8, 2011.

[STA+06] M. Sharon, T. Taverner, X. I. Ambroggio, R. J. Deshaies, and C. V. Robinson. Structural organization of the 19s proteasome lid: insights from ms of intact complexes. *PLoS biology*, 4(8):e267, 2006.

[Str05] K. Strupat. Molecular weight determination of peptides and proteins by esi and maldi. *Methods in enzymology*, 405:1, 2005.

[SXK+13] A. Schmidt, H. Xu, A. Khan, T. O'Donnell, S. Khurana, L. King, J. Manischewitz, H. Golding, P. Suphaphiphat, A. Carfi, E. Settembre, P. Dormitzer, T. Kepler, R. Zhang, A. Moody, B. Haynes, H-X. Liao, D. Shaw, and S. Harrison. Preconfiguration of the antigen-binding site during affinity maturation of a broadly neutralizing influenza virus antibody. *PNAS*, 110(1):264–269, 2013.

[SZZ07] Igor N Serdyuk, Nathan R Zaccai, and Joseph Zaccai. *Methods in molecular biophysics: structure, dynamics, function*. Cambridge University Press, 2007.

[Tar83] R. E. Tarjan. *Data Structures and Network Algorithms*, volume 44 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1983.

[THS+08] T. Taverner, H. Hernández, M. Sharon, B.T. Ruotolo, D. Matak-Vinkovic, D. Devos, R.B. Russell, and C.V. Robinson. Subunit architecture of intact protein complexes from mass spectrometry and homology modeling. *Accounts of chemical research*, 41(5):617–627, 2008.

[TRT+10] Brian Turner, Sabry Razick, Andrei L Turinsky, James Vlasblom, Edgard K Crowdy, Emerson Cho, Kyle Morrison, Ian M Donaldson, and Shoshana J Wodak. irefweb: interactive analysis of consolidated protein interaction data and their supporting evidence. *Database*, 2010:baq023, 2010.

[Vaz01] V. V. Vazirani. *Approximation Algorithms*. Springer-Verlag New York, Inc., New York, NY, USA, 2001.

[vdHvDM+06] Robert HH van den Heuvel, Esther van Duijn, Hortense Mazon, Silvia A Synowsky, Kristina Lorenzen, Cees Versluis, Stan JJ Brouns, Dave Langridge, John van der Oost, John Hoyes, et al. Improving the performance of a quadrupole time-of-flight instrument for macromolecular mass spectrometry. *Analytical chemistry*, 78(21):7473–7483, 2006.

[WR10] S.R. Wente and M.P. Rout. The nuclear pore complex and nuclear transport. *Colde Spring Harbor Perspectives in Biology*, 2(10):a000562, 2010.

[YF84] Masamichi Yamashita and John B Fenn. Electrospray ion source. another variation on the free-jet theme. *The Journal of Physical Chemistry*, 88(20):4451–4459, 1984.

[YMJ12] J. Yu, T. Murali, and R.L. Finley Jr. Assigning confidence scores to protein–protein interactions. In *Two Hybrid Technologies*, pages 161–174. Springer, 2012.

[YWZ+12] Bing Yang, Yan-Jie Wu, Ming Zhu, Sheng-Bo Fan, Jinzhong Lin, Kun Zhang, Shuang Li, Hao Chi, Yu-Xin Li, Hai-Feng Chen, et al. Identification of cross-linked peptides from complex samples. *Nature methods*, 9(9):904–906, 2012.

[ZCGB13] H. Zhang, W. Cui, M. L. Gross, and R. E. Blankenship. Native mass spectrometry of photosynthetic pigment-protein complexes. *FEBS Letters*, 2013.

[ZMB+11] M. Zhou, N. Morgner, N. P. Barrera, A. Politis, S. C. Isaacson, D. Matak-Vinković, T. Murata, R. A. Bernal, D. Stock, and C. V. Robinson. Mass spectrometry of intact v-type atpases reveals bound lipids and the effects of nucleotide binding. *Science*, 334(6054):380–385, 2011.

[ZSF+08] Min Zhou, Alan M. Sandercock, Christopher S. Fraser, Gabriela Ridlova, Elaine Stephens, Matthew R. Schenauer, Theresa Yokoi-Fong, Daniel Barsky, Julie A. Leary, John W. Hershey, Jennifer A. Doudna, and Carol V. Robinson. Mass spectrometry reveals modularity and a complete subunit interaction map of the eukaryotic translation factor eif3. *PNAS*, 105(47):18139–18144, 2008.

Chapter 8

Supplemental: Biological Systems Studied

8.1 Supplemental: Lists of Oligomers for the Assemblies Studied

In this section, we list the composition of the **complexes**, also called **oligomers**, produced experimentally and used as input of our connectivity inference problems.

8.1.1 Yeast Exosome

The 19 oligomers generated using tandem mass spectrometry and subdenaturing concentration of organic solvents [HDT+06] are the following ones:

```
List of proteins
Csl4 Dis3 Mtr3 Rrp4 Rrp40 Rrp41 Rrp42 Rrp43 Rrp45 Rrp46

List of oligomers
Mtr3 Rrp42
Rrp41 Rrp45
Rrp43 Rrp46
Rrp40 Rrp45 Rrp46
Rrp4 Rrp41 Rrp42 Rrp45
Rrp40 Rrp43 Rrp45 Rrp46
Dis3 Rrp4 Rrp41 Rrp42 Rrp45
Mtr3 Rrp4 Rrp41 Rrp42 Rrp45
Dis3 Mtr3 Rrp4 Rrp41 Rrp42 Rrp45
Dis3 Rrp4 Rrp40 Rrp41 Rrp42 Rrp45 Rrp46
Dis3 Mtr3 Rrp4 Rrp40 Rrp41 Rrp42 Rrp43 Rrp45
Dis3 Mtr3 Rrp4 Rrp40 Rrp41 Rrp42 Rrp45 Rrp46
Dis3 Mtr3 Rrp4 Rrp41 Rrp42 Rrp43 Rrp45 Rrp46
Dis3 Mtr3 Rrp40 Rrp41 Rrp42 Rrp43 Rrp45 Rrp46
Mtr3 Rrp4 Rrp40 Rrp41 Rrp42 Rrp43 Rrp45 Rrp46
Csl4 Dis3 Mtr3 Rrp4 Rrp41 Rrp42 Rrp43 Rrp45 Rrp46
Csl4 Dis3 Mtr3 Rrp40 Rrp41 Rrp42 Rrp43 Rrp45 Rrp46
Csl4 Mtr3 Rrp4 Rrp40 Rrp41 Rrp42 Rrp43 Rrp45 Rrp46
Dis3 Mtr3 Rrp4 Rrp40 Rrp41 Rrp42 Rrp43 Rrp45 Rrp46
```

8.1.2 Yeast 19S Proteasome lid

The 14 oligomers obtained using MS and MS/MS (8 of them), and cross-linking experiments (6 of them) are as follows [STA+06]:

```
List of proteins
Rpn3 Rpn5 Rpn6 Rpn7 Rpn8 Rpn9 Rpn11 Rpn12 Sem1

List of oligomers
Rpn7 Sem1
Rpn3 Sem1
Rpn3 Rpn5
Rpn3 Rpn5 Rpn8
Rpn5 Rpn6 Rpn8
Rpn5 Rpn8 Rpn9
Rpn6 Rpn8 Rpn9
Rpn3 Rpn5 Rpn8 Rpn9
Rpn5 Rpn6 Rpn8 Rpn9
Rpn3 Rpn5 Rpn7 Rpn9 Rpn11
Rpn3 Rpn5 Rpn6 Rpn7 Rpn8 Rpn11 Sem1
Rpn3 Rpn5 Rpn6 Rpn7 Rpn8 Rpn9 Rpn11 Sem1
Rpn3 Rpn5 Rpn6 Rpn7 Rpn8 Rpn11 Rpn12 Sem1
Rpn3 Rpn5 Rpn7 Rpn8 Rpn9 Rpn11 Rpn12 Sem1
```

8.1.3 Eukaryotic Translation factor eIF3

The 27 oligomers obtained using tandem mass spectrometry [ZSF+08] are the following ones:

```
List of proteins
a b c d e f g h i k l m

List of oligomers
b g
d e
e l
f h
f m
g i
h m
k l
b g i
d e l
e k l
f h m
c d e l
c e k l
d e k l
a b c g i
a b e g i
c d e h l
c d e k l
c d e f h l
c d e h k l
```

```
c d e f h k l
c d e f h l m
c d e g i k l
c d e f h k l m
b c d e f g h i m
b c d e f g h i k l m
```

Note that the subunit eIF3j is excluded from the list of protein types as there is no sub-complex (oligomer) yielded containing the same.

8.2 Supplemental: Reference Contacts Within Assemblies

In this section, we provide a classification of various contacts reported in the literature, classified as a function of the experimental technique they were observed with. These contact categories are used to define reference edge sets used for the assessment of the edges reported by our algorithms.

8.2.1 Pairwise Contacts within Macro-molecular Oligomers

Crystal contacts: [$\mathbf{C_{Xtal}}$] A high-resolution crystal structure of an assembly can be seen as the gold standard providing all pairwise contacts between its constituting molecules. Given such a crystal structure, all pairs of molecules are tested to check whether they define a contact. A pair defines a contact provided that in the solvent accessible (SAS) model of the assembly [1], two atoms from these pairs define an edge in the α-complex of the assembly for $\alpha = 0$, as classically done to define macro-molecular interfaces [CPBJ06, JBC08, LC10].)

This protocol actually calls for one comment. For protein interfaces, it is generally accepted that any biologically specific contact has a surface area beyond 500Å^2, or equivalently, involves at least 50 atoms on each partners [JBC08]. For assemblies, because of the promiscuity of molecules, this threshold does not apply directly. As an example, consider the number of atoms observed at interfaces for the yeast exosome (Table 8.1). While selected interfaces meet the usual criterion, others involve a handful of atoms. For this reason, in addition to C_{Xtal}, we defined a set C^{-}_{Xtal} involving the most prominent contacts only (14 contacts out of 26). We note in passing that the existence of a hierarchy of interface size within a protein assembly has been reported in [LBERT08, JBC08].

Cryo-electron microscopy reconstruction (set $\mathbf{C_{Cryo}}$). Cryo-electron microscopy is a technique to visualize and interpret unstained biological samples including macromolecular assemblies of 200 kDa and more. The biological sample is cryo-fixed to preserve the aqueous environment around the macromolecule thereby, preventing ultrastructural changes, redistribution of elements etc. The imaging is therefore done in near native conditions and using state of art computer controlled microscopy, image reconstruction software, sub-nanometer resolution structures of large biological macromolecular assemblies can be retrieved.

Cross-linking (set $\mathbf{C_{XL}}$). Cross-linking is an analytical technique which consists in chemically linking surface residue of two proteins located nearby. This technique is used to identify protein-protein interactions, upon disrupting the cell and identifying the cross-linked proteins. The outcome allows identifying interacting proteins within an assembly, but also transient interactions which get stabilized by the cross-linker. The distance between the two amino-acids cross-linked is circa 25Å, including the length of the linker and the span of the side-chains of the two amino-acids involved.

Due to this distance, the two proteins cross-linked may not form an interface in the sense defined above. However, cross-linking contacts are considered as interfacial contacts in [KRY+12], defining a *low-resolution topology*.

[1] Given a van der Waals model, the corresponding SAS model consists of expanding the atomic radii by 1.4Å, so as to account for an implicit layer of water molecules on the model. The SAS model also allows capturing intersections between atoms which are nearby in 3D space, but are not covalently bonded.

Dimers obtained from various biophysical experiments (set $\mathbf{C_{Dim}}$). The following experiments deliver information on the existence of a dimer involving two proteins:

- Mass spectrometry (MS) or Tandem Mass spectrometry (MS/MS): upon collecting a dimer, and since no re-arrangement occurs in gas phase, the two proteins form a dimer in the assembly analyzed.

- Tandem affinity purification (TAP): a bait put on one protein pulls down another protein, upon capturing the marked protein on a affinity purification column.

- Co-immuno-precipitation of two proteins: as above.

- Native Agarose Gel electrophoresis: two proteins are inferred to be interacting if instead of two sharp bands (assuming mol. wt. to be different) a broad band spread over a range of molecular weight is observed.

- NMR titrations: information of the interacting residues of one protein is inferred from the perturbation of the chemical shifts of the interfacial residues obtained when adding the partner.

8.2.2 Yeast Exosome

C_{Xtal} (26 contacts)			C_{Dim} (7 contacts)
X-Ray Crystallography, 2.8 Å [MBC13]			TAP, MS, MS/MS
Chains	Subunits	#Interface atoms	*Partial Denaturation* [HDT+06] [THS+08]
CG	(Rrp43, Rrp40)	2	(Rrp43, Csl4)
EI	(Rrp42, Csl4)	6	(Rrp45, Rrp40)
AF	(Rrp45, Mtr3)	19	(Rrp46, Rrp40)
FH	(Mtr3, Rrp4)	24	(Rrp45, Rrp46)
DF	(Rrp46, Mtr3)	54	(Rrp45, Rrp41)
AH	(Rrp45, Rrp4)	59	(Rrp43, Rrp46)
HI	(Rrp4, Csl4)	60	(Rrp42, Mtr3)
AC	(Rrp45, Rrp43)	72	
DI	(Rrp46, Csl4)	79	
AJ	(Rrp45, Dis3)	95	
GI	(Rrp40, Csl4)	117	
CJ	(Rrp43, Dis3)	148	
CI	(Rrp43, Csl4)†	211	
BE	(Rrp41, Rrp42)	223	
EJ	(Rrp42, Dis3)	231	
AG	(Rrp45, Rrp40)†	245	
EF	(Rrp42, Mtr3)†	313	
FI	(Mtr3, Csl4)	327	
AD	(Rrp45, Rrp46)†	349	
BH	(Rrp41, Rrp4)	352	
CD	(Rrp43, Rrp46)†	369	
BJ	(Rrp41, Dis3)	371	
DG	(Rrp46, Rrp40)†	411	
CF	(Rrp43, Mtr3)	446	
EH	(Rrp42, Rrp4)	458	
AB	(Rrp45, Rrp41)†	463	

† signifies those contacts which are also recovered by other biophysical experiments, TAP, MS, MS/MS

Table 8.1: **List of contacts determined from experiments for Yeast Exosome.** Note in particular the crystal contacts (third column), determined from the crystal structure using the relative atomic positions, using a Voronoi based interface model [LC10]. In general, interfaces involving a large number of atoms are stable ones. Note also that small interfaces in the final product may correspond to interfaces which were large at an early stage of the assembly formation, and which got shrunk along the accretion of molecules.

Published previously in the panel C of Fig. 4 of [THS+08] a list of the contacts determined using *Network inference* algorithm for the set of oligomers for Yeast exosome in the Section 8.1. They are as follows:

```
12 contacts
(Csl4, Rrp43)
(Dis3, Rrp45)
(Mtr3, Rrp42)
(Mtr3, Rrp43)
(Rrp4, Rrp41)
(Rrp4, Rrp42)
(Rrp40, Rrp45)
(Rrp40, Rrp46)
```

```
(Rrp41, Rrp42)
(Rrp41, Rrp45)
(Rrp43, Rrp46)
(Rrp45, Rrp46)
```

8.2.3 Yeast Proteasome Lid

C_{Cryo} (13 contacts)	C_{Dim} (3 contacts)	C_{XL} (14 contacts)	
[LEM+12]	MS, MS/MS analysis [STA+06]	$CX-DSSO, DSS, BS3$	$References$
(Rpn3, Rpn5)†	(Rpn5, Rpn8)	(Rpn3, Rpn7)	[KRY+12][LFB+12]
(Rpn3, Rpn8)†	(Rpn6, Rpn8)	(Rpn3, Rpn8)	[KRY+12]
(Rpn3, Rpn12)†	(Rpn8, Rpn9)	(Rpn3, Rpn12)	[KRY+12]
(Rpn5, Rpn6)†		(Rpn3, Sem1)	[KRY+12][STA+06]
(Rpn5, Rpn8)†		(Rpn5, Rpn6)	[KRY+12]
(Rpn5, Rpn9)†		(Rpn5, Rpn9)	[KRY+12][LFB+12]
(Rpn5, Rpn11)		(Rpn6, Rpn7)	[KRY+12]
(Rpn6, Rpn7)†		(Rpn6, Rpn11)	[KRY+12]
(Rpn6, Rpn11)†		(Rpn7, Rpn11)	[KRY+12]
(Rpn7, Rpn8)		(Rpn7, Sem1)	[KRY+12][STA+06]
(Rpn8, Rpn9)†		(Rpn8, Rpn9)	[KRY+12]
(Rpn8 Rpn11)†		(Rpn8, Rpn11)	[KRY+12]
(Rpn9, Rpn11)		(Rpn3, Rpn5)	[STA+06]
		(Rpn3, Rpn11)	[LFB+12]

† signifies those contacts in C_{Cryo} which are also recovered by other biophysical experiments, TAP, MS, MS/MS, *cross-links*

Number of distinct contacts, $|C_{Cryo} \cup C_{Dim} \cup C_{XL}| = 19$

Table 8.2: **List of contacts determined from experiments for Yeast 19S Proteasome Lid**

Published previously in the panel B of Fig. 3 of [THS+08], a list of the contacts determined using *Network inference* algorithm for the set of oligomers for Yeast 19S Proteasome lid in the section 8.1. They are as follows:

```
9 contacts
(Rpn3, Rpn5)
(Rpn3, Rpn11)
(Rpn3, Sem1)
(Rpn5, Rpn7)
(Rpn5, Rpn8)
(Rpn5, Rpn11)
(Rpn6, Rpn8)
(Rpn7, Sem1)
(Rpn8, Rpn9)
```

8.2.4 eIF3

C_{Cryo} (15 contacts)[QASV$^+$13]	C_{Dim} (10 contacts)[ZSF$^+$08]
(a, c)	(a, b)
(a, d)	(b, g)
(a, m)	(b, i)
(b, c)	(d, e)
(b, g)†	(e, l)
(b, i)†	(f, h)
(c, e)	(f, m)
(c, h)	(g, i)
(e, l)†	(h, m)
(f, h)†	(k, l)
(f, m)†	
(g, i)†	
(h, m)†	
(h, l)	
(k, l)†	

† signifies those contacts in C_{Cryo} which are also recovered by other biophysical experiments, TAP, MS, MS/MS
Number of distinct contacts, $|C_{Cryo} \cup C_{Dim}| = 17$

Table 8.3: **List of contacts determined from experiments for eIF3**.

Published previously in Fig. 4 of [ZSF$^+$08], a list of the contacts determined manually for the set of oligomers for eIF3 in the section 8.1. They are as follows:

```
17 contacts
(a,b)
(a,c)
(b,c)
(b,e)
(b,f)
(b,g)
(b,i)
(c,d)
(c,e)
(c,h)
(d,e)
(e,l)
(f,h)
(f,m)
(g,i)
(h,m)
(k,l)
```

Topics in Mass Spectrometry Based Structure Determination
English summary

Mass spectrometry (MS), an analytical technique initially invented to deal with small molecules, has emerged over the past decade as a key approach in structural biology. The recent advances have made it possible to transfer large macromolecular assemblies into the vacuum without their dissociation, raising challenging algorithmic problems. This thesis makes contributions to three such problems.

The first contribution deals with *stoichiometry determination* (SD), namely the problem of determining the number of copies of each subunit of an assembly, from mass measurements. We deal with the *interval SD problem*, where the target mass belongs to an interval accounting for mass measurement uncertainties. We present a constant memory space algorithm (`DIOPHANTINE`), and an output sensitive dynamic programming based algorithm (`DP++`), outperforming state-of-the-art methods both for integer type and float type problems.

The second contribution deals with the inference of pairwise contacts between subunits, using a list of sub-complexes whose composition is known. We introduce the *Minimum Connectivity Inference* problem (MCI) and present two algorithms solving it. We also show an excellent agreement between the contacts reported by these algorithms and those determined experimentally.

The third contribution deals with *Minimum Weight Connectivity Inference* (MWCI), a problem where weights on candidate edges are available, reflecting their likelihood. We present in particular a bootstrap algorithm allowing one to report a set of edges with improved sensitivity and specificity with respect to those obtaining upon solving MCI.

Sur quelques problèmes algorithmiques relatifs à la détermination de structure à partir de données de spectrométrie de masse
Résumé Français

La spectrométrie de masse, initialement développée pour de petites molécules, a permis au cours de la dernière écoulée d'étudier en phase gazeuse des assemblages macro-moléculaires intacts, posant nombre de questions algorithmiques difficiles, dont trois sont étudiées dans cette thèse.

La première contribution concerne la *détermination de stoichiometrie* (SD), et vise à trouver le nombre de copies de chaque constituant dans un assemblage. On étudie le cas où la masse cible se trouve dans un intervalle dont les bornes rendent compte des incertitudes des mesures des masses. Nous présentons un algorithme de taille mémoire constante (`DIOPHANTINE`), et un algorithme de complexité sensible à la sortie (`DP++`), plus performants que l'état de l'art, pour des masses en nombre entier ou flottant.

La seconde contribution traite de l'inférence de connectivité à partir d'une liste d'oligomères dont la composition en termes de sous-unités est connue. On introduit le problème d'*inférence de connectivité minimale* (MCI) et présente deux algorithmes pour le résoudre. On montre aussi un accord excellent entre les contacts trouvés et ceux déterminés expérimentalement.

La troisième contribution aborde le problème d'*inference de connectivité de poids minimal*, lorsque chaque contact potentiel a un poids reflétant sa probabilité d'occurrence. On présente en particulier un algorithme de bootstrap permettant de trouver un ensemble d'arêtes de *sensitivité* et *specificité* meilleures que celles obtenues pour les solutions du problème MCI.