Appendix A

Zusammenfassung

Neue Forschungsergebnisse zu proteomischen Techniken, zum Beispiel Two-Hybrid und Biochemical Purification, erlauben Untersuchungen von Protein-Interaktionen in großem Maßstab. Diese Arbeit untersucht modellbasierte Ansätze, um aus Tandem-Affinity-Purification-Experimenten Proteinkomplexe zu berechnen. Wir vergleichen ein einfaches Modell, dass Überlappungen zwischen Komplexen zulässt, mit einem Partitionsmodell. Außerdem stellen wir ein Visualisierungsverfahren vor, dass überlappende Komplexe in experimentell ermittelten Daten darstellt.

Bisherige Techniken zur Analyse von Proteininteraktionen stützen sich auf Heuristiken. Sie produzieren nützliche Ergebnisse, aber legen kein an der Natur orientiertes Modell für die Bildung von Protein-Komplexen zugrunde. Außerdem benötigen haben diese Verfahren oft eine Vielzahl von Parametern, die für den Nutzer nur schwierig korrekt einzustellen sind. Modellbasierte Techniken erlauben eine prinzipiellere Betrachtung von Proteinanalyse, weil sie explizit und quantitativ beschreiben, wie Proteininteraktionen in der Natur sich in experimentellen Ergebnissen niederschlagen. Unser Algorithmus berechnet dann auf der Basis dieses Modells die wahrscheinlichsten Proteinkomplexe.

Wir schlagen zwei Modelle vor, um Proteinkomplexe zu berechnen. Das erste, in gewissem Sinne einfachst mögliche, basiert auf Frequent Itemset Mining und zählt das Auftreten von Mengen von Proteinen in den experimentellen Ergebnissen. Wir neben dabei an, dass die Neigung von Proteinen, bestimmte Komplexe zu bilden, für unterschiedliche Komplexe statistisch unabhängig ist, insbesondere auch dann, wenn die gleichen Proteine an den Komplexen beteiligt sind. Komplexe, die einander überlappen, sind damit erlaubt. Das zweite Modell stellt das andere Extrem dar und nimmt an, dass Komplexe die Menge von Proteinen partitionieren. Komplexe damit einander nicht überlappen können und sich Komplexbildung auf rein paarweises Verhalten von Proteinen zurückführen lässt. In diesem Modell ist die Beobachtung einer Interaktion zwischen einem Proteinpaar wahrscheinlicher, wenn beide Proteine miteinander in einem Komplex vorkommen. Beruhend auf diesem Modell nutzen wir Markov Random Fields, um eine Maximum-Likelihood-Schätzung von Komplexen zu berechnen.

Wir vergleichen die Effektivität beider Modelle anhand von Benchmarks. Der Vergleich ergab, dass das Partitionsmodell sehr viel besser funktioniert als das Modell mit überlappenden Komplexen, was darauf hinweist, dass Komplexbildung in der Natur im Großsen und Ganzen ein paarweises Phänomen ist, auch wenn einige Gegenbeispiele dokumentiert sind. Die Leistung unseres Modells ist mindestens ebenso gut wie bisherige Techniken, die auf Heuristiken basieren, hat aber den Vorteil, dass es ohne einzustellende Parameter auskommt. Wir sind deshalb zuversichtlich, dass

das Modell auch auf unbekannten Datensätzen gut funktioniert.

Abschließend entwickeln wir ein nützliche Visualisierungsverfahren für Tandem-Affinity-Purification-Daten. Die Ergebnisse von Purification werden als ein gerichteter Graph modelliert. Die Kanten modellieren die Wahrscheinlichkeit, dass eine Purification die Untermenge der anderen ist. Dieses Verfahren berücksichtigt damit die assymmetrische Beziehung zwischen zwei Experimenten. Wir zeigen die Effektivität des Verfahrens anhand einer Visualisierung einer der neusten Purifications.

Appendix B

Software for MRF

We implemented the algorithm in C++, extensively using the standard template library (STL). We use the GNU g++ compiler. The source code (including the Makefile) is available at `http://algorithmics.molgen.mpg.de/ProteinComplexes`. The software has been tested on the Linux operating system and MAC OS. To compile the program, simply run make.

Given a user-defined number of cluster and input parameter ψ, the program estimates the cluster assignment for each protein directly from the input purification. To select a number of cluster, this program must be run with different number of clusters and select the output with the maximum likelihood. We suggest running the search in a distributed computing environment with at least 30 processors. Each processor should have at least 512 MB of memory.

B.1 Usage

```
Usage: estimate [-m] purification  K psi output-file
```

Option:

- **m**: when present, select the matrix model (default: without the option, select spoke model).

Arguments:

- **purification**: input purification file (a comma-separated file).
- **K**: the number of clusters.

- **psi** (ψ): the initial log ratio of the error rate. We recommend values between 2.0-5.0 for the spoke model and 7.0-10.0 for the matrix model.

- **output-file**: the name of the output file listing the cluster assignment for each protein.

B.2 Input file format

The input file should contain all purifications from a single experiment. It must be a comma-separated file. Each line contains all proteins (a bait and preys) from one purification. The first protein must be the bait protein. The order of prey proteins is not important.

Example:

```
YBR128C,YPL120W
YKL135C,YHL019C,YLR170C,YPL259C,YPR010C,YPR029C
YBL037W,YJR058C,YOL062C
YPL195W,YGL019W,YGR261C,YJL024C
...
```

B.3 Output file format

The first four lines of the output file contain the likelihood and error rates used to compute the likelihood. Each subsequent line is a cluster assignment for each protein present in the input file.

Example:

```
Lambda  178364.73092 (the likelihood)
FN      0.75240 (the false negative rate)
FP      0.00169 (the false positive rate)
Constant        17.62524 (the estimated log ratio of the error rate)
YBR236C 105
YOR151C 259
YNR016C 185
```

```
YJR064W 48
YLR386W 67
YAR015W 266
...
```

Declaration

I declare that this thesis contains my original work. I have written this thesis myself and did not copy from other sources, publications or previously published thesis.

Berlin, 14.06.2007

Wasinee Rungsarityotin