CS 466
Introduction to Bioinformatics
Lecture 14

Mohammed El-Kebir

October 11, 2019

Outline

- Recap: RNA Secondary Structure Prediction
- Protein Contact Map Overlap

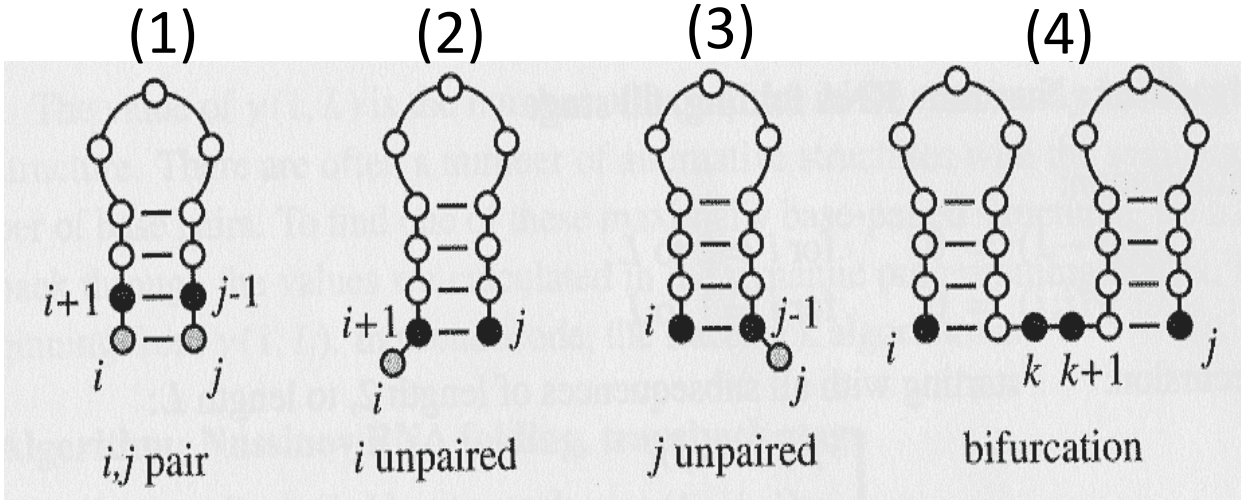
Reading:

- Lecture notes
- Caprara, A., Carr, R., Istrail, S., Lancia, G., & Walenz, B. (2004). 1001 Optimal PDB Structure Alignments: Integer Programming Methods for Finding the Maximum Contact Map Overlap. *Journal of Computational Biology*, *11*(1), 27–52. http://doi.org/10.1089/10665270477341687 6

Nussinov Algorithm – Dynamic Programming

Problem: Given RNA sequence $\mathbf{v} \in \{\mathrm{A}, \mathrm{U}, \mathrm{C}, \mathrm{G}\}^n$, find a *pseudoknot-free secondary structure* with the maximum number of complementary base pairings

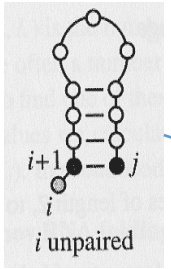
Let $s[i,j]$ denote the maximum number of pseudoknot-free complementary base pairings in subsequence $v_i, \ldots, v_j$

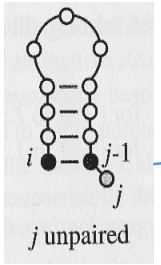


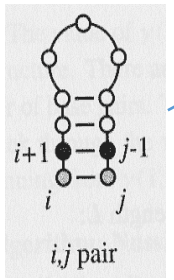
$$s[i,j] = \max \begin{cases} 0, & \text{if } i \geq j, \\ s[i+1, j-1] + 1, & \text{if } i < j \text{ and } (v_i, v_j) \in \Gamma, \quad (1) \\ s[i+1, j-1], & \text{if } i < j \text{ and } (v_i, v_j) \notin \Gamma, \quad (1^*) \\ s[i+1, j], & \text{if } i < j, \quad (2) \\ s[i, j-1], & \text{if } i < j, \quad (3) \\ \max_{i<k<j}\{s[i,k] + s[k+1, j]\}, & \text{if } i < j, \quad (4) \end{cases}$$

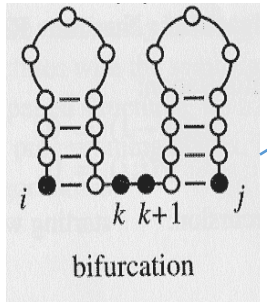
Question: Which case is redundant?

Nussinov Algorithm – Traceback Step



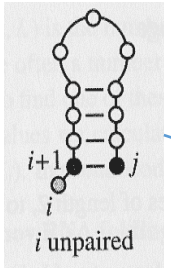


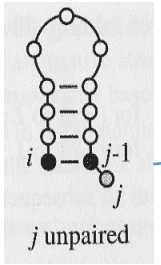


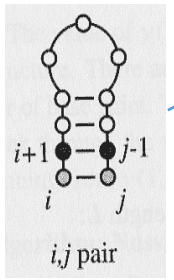


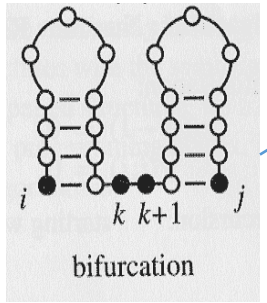
Push (1, *n*) onto stack
Repeat until stack is empty:
pop (*i*,*j*)
if *i* ≥ *j* continue
else if s[*i*+1,*j*] = s[*i*,*j*]
push (*i*+1,*j*)
else if s[*i*,*j*-1] = S[*i*,*j*]
push (i,j-1)
else if s[i+1,j-1] + 1 = s[i,j]
record (i,j) base pair
push (i+1,j-1)
else for k = i+1 to j-1
if s[i,k]+s[k+1,j] = s[i,j]
push (k+1,j)
push (i,k)
break (for loop)

Nussinov Algorithm – Traceback Step









Push (1, *n*) onto stack
Repeat until stack is empty:
pop (*i*,*j*)
if *i* ≥ *j* continue
else if s[*i*+1,*j*] = s[*i*,*j*]
push (*i*+1,*j*)
else if s[*i*,*j*-1] = S[*i*,*j*]
push (i,j-1)
else if s[i+1,j-1] + 1 = s[i,j]
record (i,j) base pair
push (i+1,j-1)
else for k = i+1 to j-1
if s[i,k]+s[k+1,j] = s[i,j]
push (k+1,j)
push (i,k)
break (for loop)

BackTrack(*i*, *j*)
if *i* < *j*
if s[*i*+1, *j*] = s[*i*, *j*]
BackTrack(*i*+1, *j*)
else if s[*i*, *j*-1] = S[*i*, *j*]
BackTrack(*i*, *j*-1)
else if s[*i*+1,*j*-1] + 1 = s[*i*, *j*]
Output (*i*, *j*)
BackTrack(*i*+1, *j*-1)
else for *k* = *i*+1 **to** *j*-1
if s[*i*, *k*]+s[*k*+1, *j*] = s[*i*, *j*]
BackTrack(*k*+1, *j*)
BackTrack(*i*, *k*)
break (for loop)

Outline

- Recap: RNA Secondary Structure Prediction
- Protein Contact Map Overlap

Reading:

- Lecture notes
- Caprara, A., Carr, R., Istrail, S., Lancia, G., & Walenz, B. (2004). 1001 Optimal PDB Structure Alignments: Integer Programming Methods for Finding the Maximum Contact Map Overlap. *Journal of Computational Biology*, *11*(1), 27–52. http://doi.org/10.1089/10665270477341687 6

Central Dogma of Molecular Biology

Three fundamental molecules:

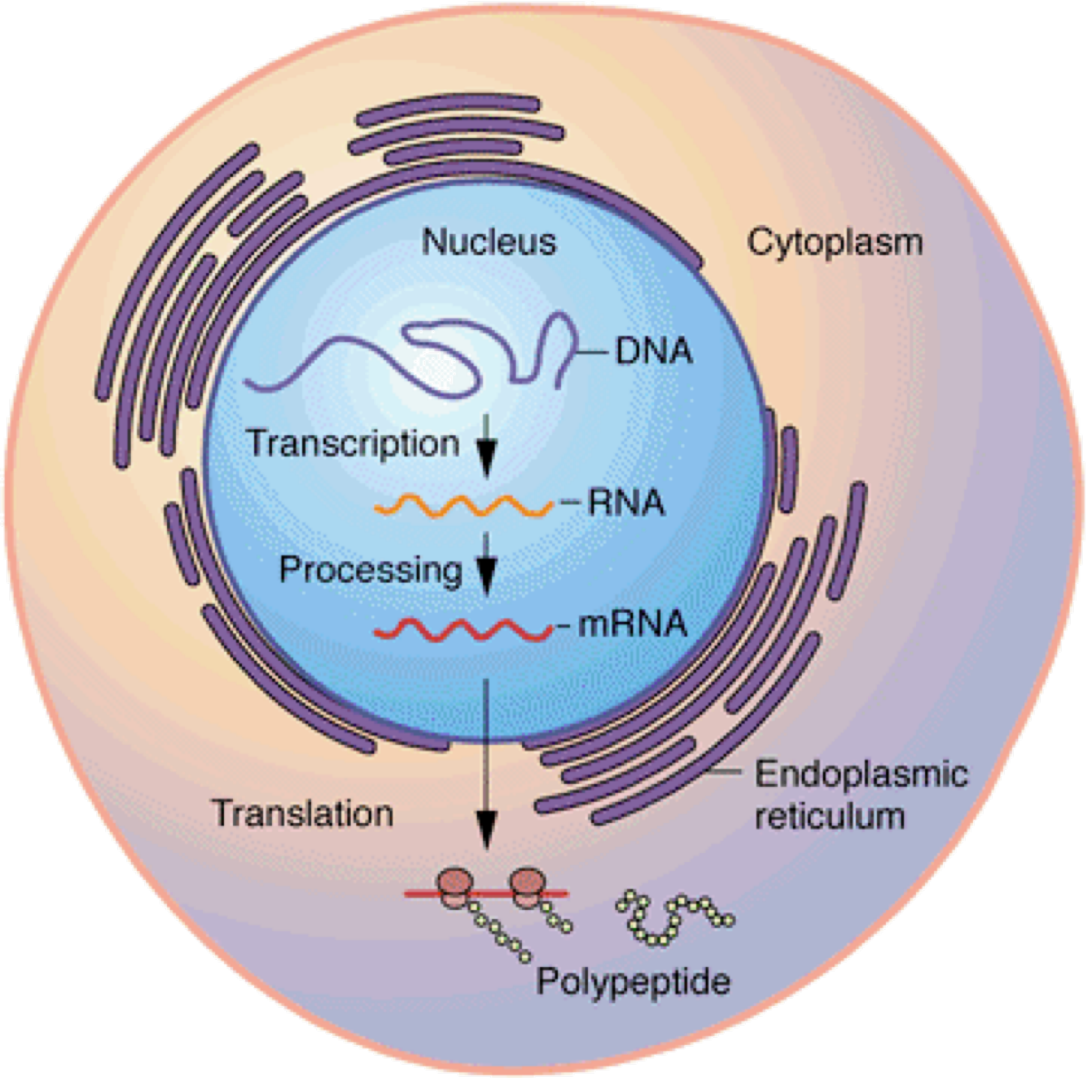
1. **DNA**
   Information storage.

2. **RNA**
   Old view: Mostly a "messenger".
   New view: Performs many important functions, through **3-D structure**!

3. **Protein**
   Perform most cellular functions (biochemistry, signaling, control, etc.)

DNA → RNA → Protein

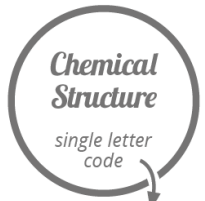
First proposed by Francis Crick in 1956.



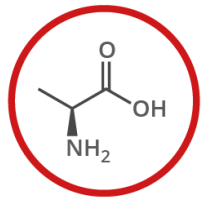
A GUIDE TO THE TWENTY COMMON AMINO ACIDS

AMINO ACIDS ARE THE BUILDING BLOCKS OF PROTEINS IN LIVING ORGANISMS. THERE ARE OVER 500 AMINO ACIDS FOUND IN NATURE - HOWEVER, THE HUMAN GENETIC CODE ONLY DIRECTLY ENCODES 20. 'ESSENTIAL' AMINO ACIDS MUST BE OBTAINED FROM THE DIET, WHILST NON-ESSENTIAL AMINO ACIDS CAN BE SYNTHESISED IN THE BODY.

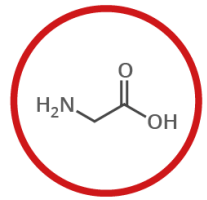
Chart Key: ALIPHATIC | AROMATIC | ACIDIC | BASIC | HYDROXYLIC | SULFUR-CONTAINING | AMIDIC | NON-ESSENTIAL | ESSENTIAL



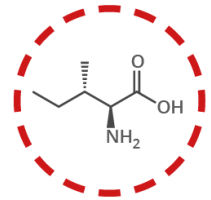
NAME **A**
three letter code
DNA codons



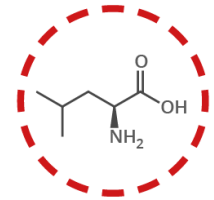
ALANINE A
Ala
GCT, GCC, GCA, GCG



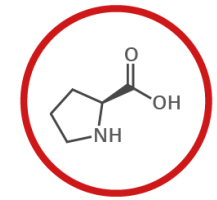
GLYCINE G
Gly
GGT, GGC, GGA, GGG



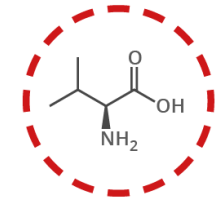
ISOLEUCINE I
Ile
ATT, ATC, ATA



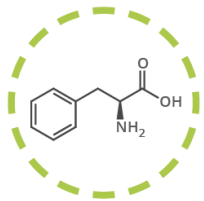
LEUCINE L
Leu
CTT, CTC, CTA, CTG, TTA, TTG



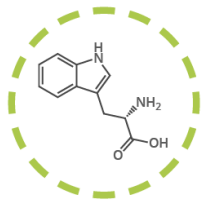
PROLINE P
Pro
CCT, CCC, CCA, CCG



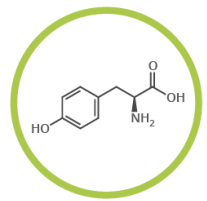
VALINE V
Val
GTT, GTC, GTA, GTG



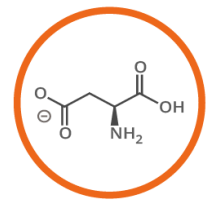
PHENYLALANINE F
Phe
TTT, TTC



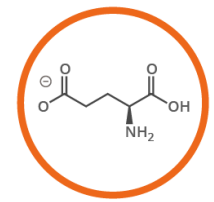
TRYPTOPHAN W
Trp
TGG



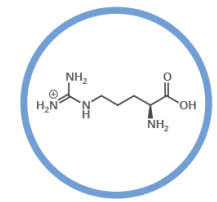
TYROSINE Y
Tyr
TAT, TAC



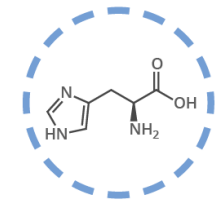
ASPARTIC ACID D
Asp
GAT, GAC



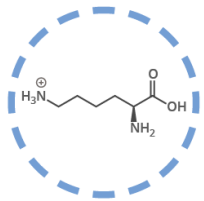
GLUTAMIC ACID E
Glu
GAA, GAG



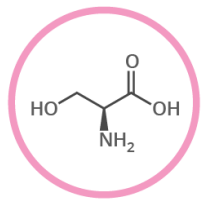
ARGININE R
Arg
CGT, CGC, CGA, CGG, AGA, AGG



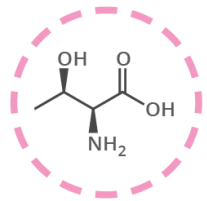
HISTIDINE H
His
CAT, CAC



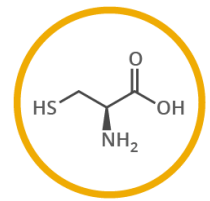
LYSINE K
Lys
AAA, AAG



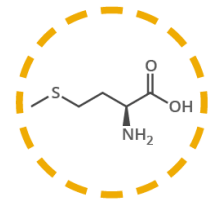
SERINE S
Ser
TCT, TCC, TCA, TCG, AGT, AGC



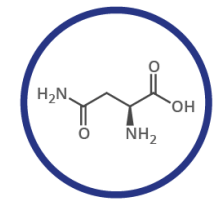
THREONINE T
Thr
ACT, ACC, ACA, ACG



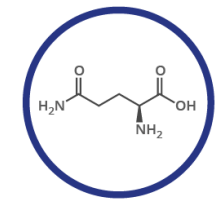
CYSTEINE C
Cys
TGT, TGC



METHIONINE M
Met
ATG



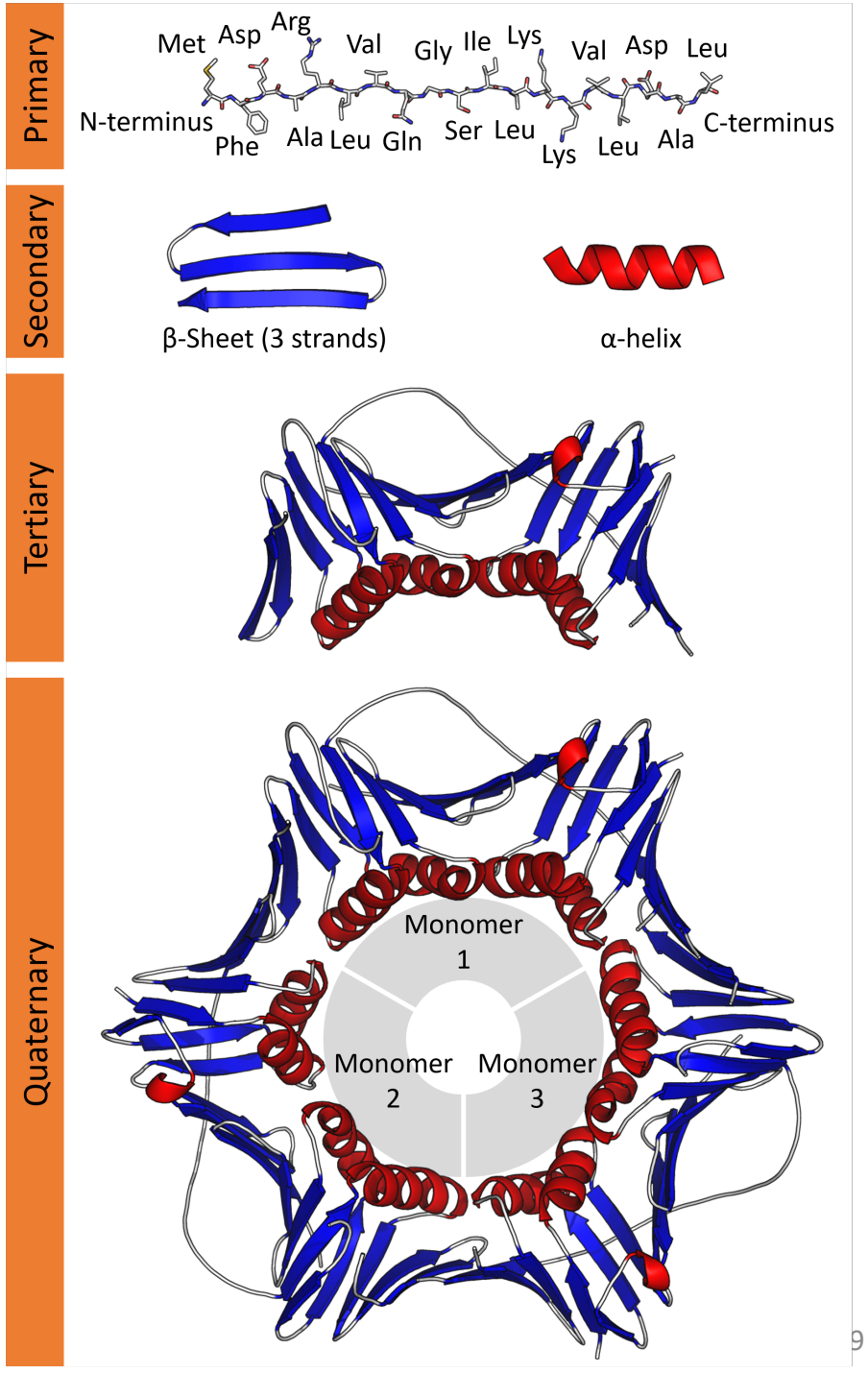
ASPARAGINE N
Asn
AAT, AAC



GLUTAMINE Q
Gln
CAA, CAG

Note: This chart only shows those amino acids for which the human genetic code directly codes for. Selenocysteine is often referred to as the 21st amino acid, but is encoded in a special manner. In some cases, distinguishing between asparagine/aspartic acid and glutamine/glutamic acid is difficult. In these cases, the codes asx (B) and glx (Z) are respectively used.

Protein Structure Prediction

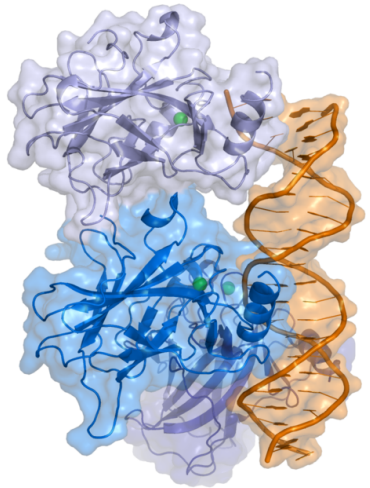


Example

- http://pdb101.rcsb.org/motm/218

On Sequence, Structure and Function: p53

```
        10         20         30         40         50
MEEPQSDPSV EPPLSQETFS DLWKLLPENN VLSPLPSQAM DDLMLSPDDI
        60         70         80         90        100
EQWFTEDPGP DEAPRMPEAA PPVAPAPAAP TPAAPAPAPS WPLSSSVPSQ
       110        120        130        140        150
KTYQGSYGFR LGFLHSGTAK SVTCTYSPAL NKMFCQLAKT CPVQLWVDST
       160        170        180        190        200
PPPGTRVRAM AIYKQSQHMT EVVRRCPHHE RCSDSDGLAP PQHLIRVEGN
       210        220        230        240        250
LRVEYLDDRN TFRHSVVVPY EPPEVGSDCT TIHYNYMCNS SCMGGMNRRP
       260        270        280        290        300
ILTIITLEDS SGNLLGRNSF EVRVCACPGR DRRTEEENLR KKGEPHHELP
       310        320        330        340        350
PGSTKRALPN NTSSSPQPKK KPLDGEYFTL QIRGRERFEM FRELNEALEL
       360        370        380        390
KDAQAGKEPG GSRAHSSHLK SKKGQSTSRH KKLMFKTEGP DSD
```

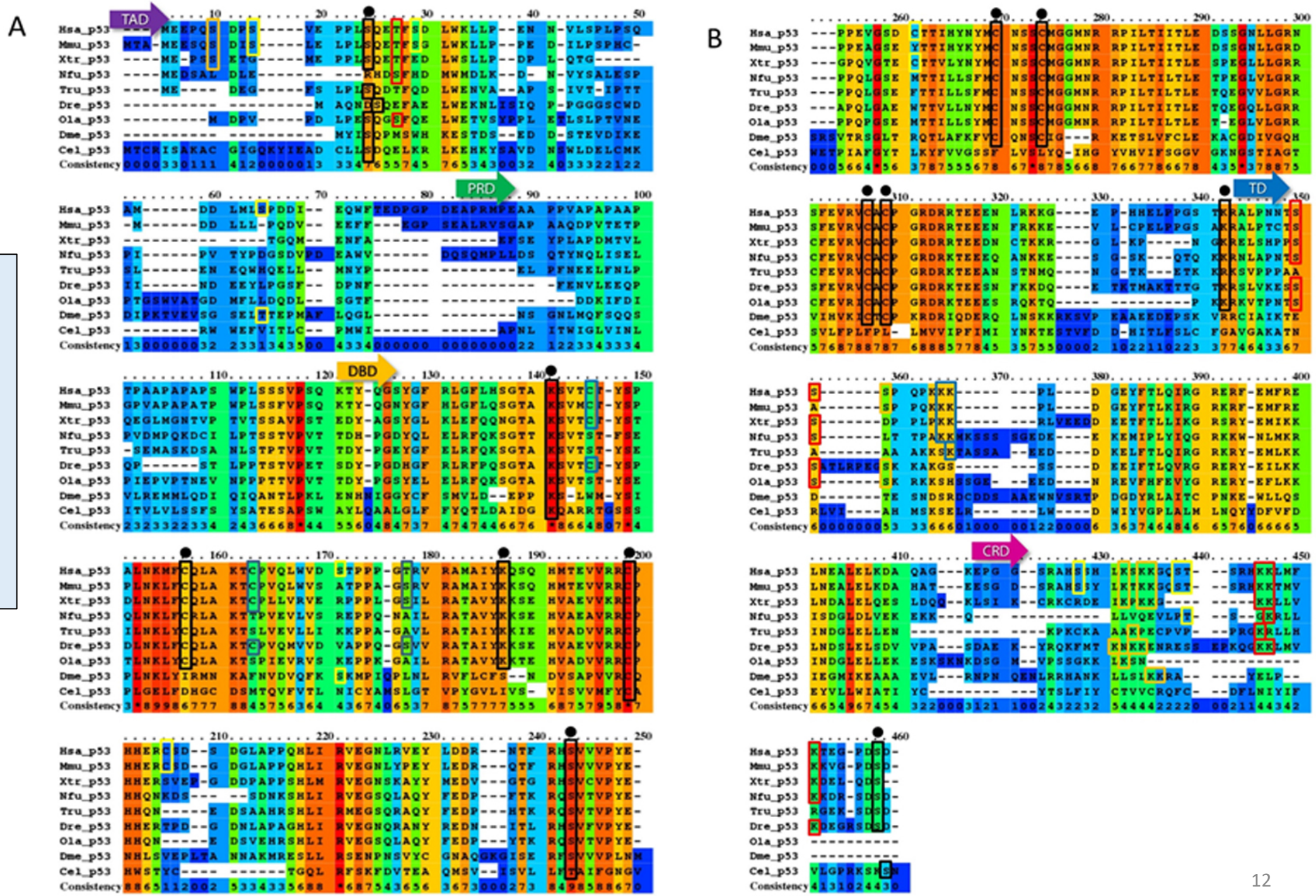


- It can activate DNA repair proteins when DNA has sustained damage
- It can arrest growth by holding the cell cycle at the G1/S regulation point on DNA damage
- It can initiate apoptosis (i.e., programmed cell death) if DNA damage proves to be irreparable.
- It is essential for the senescence response to short telomeres.

Sequence → Structure → Function

What is functionally important is conserved throughout evolution

What is functionally important is conserved throughout evolution



How to Compare Two Protein Sequences?

TP53 (Human)

```
  1 meepqsdpsv epplsqetfs dlwkllpenn vlsplpsqam ddlmlspddi eqwftedpgp
 61 deaprmpeaa ppvapapaap tpaapapaps wplsssvpsq ktyqgsygfr lgflhsgtak
121 svtctyspal nkmfcqlakt cpvqlwvdst pppgtrvram aiykqsqhmt evvrrcphhe
181 rcsdsdglap pqhlirvegn lrveylddrn tfrhsvvvpy eppevgsdct tihynymcns
241 scmggmnrrp iltiitleds sgnllgrnsf evrvcacpgr drrteeenlr kkgephhelp
301 pgstkralpn ntssspqpkk kpldgeyftl qirgrerfem frelnealel kdaqagkepg
361 gsrahsshlk skkgqstsrh kklmfktegp dsd
```

p53 (Mouse)

```
  1 mtameesqsd islelplsqe tfsglwkllp pedilpsphc mddlllpqdv eeffegpsea
 61 lrvsgapaaq dpvtetpgpv apapatpwpl ssfvpsqkty qgnygfhlgf lqsgtaksvm
121 ctyspplnkl fcqlaktcpv qlwvsatppa gsrvramaiy kksqhmtevv rrcphhercs
181 dgdglappqh rirvegnlyp eyledrqtfr hsvvvpyepp eagseyttih ykymcnsscm
241 ggmnrrpilt iitledssgn llgrdsfevr vcacpgrdrr teeenfrkke vlcpelppgs
301 akralptcts asppqkkkpl dgeyftlkir grkrfemfre lnealelkda hateesgdsr
361 ahssylktkk gqstsrhkkt mvkkvgpdsd
```

How to Compare Two Protein Sequences?

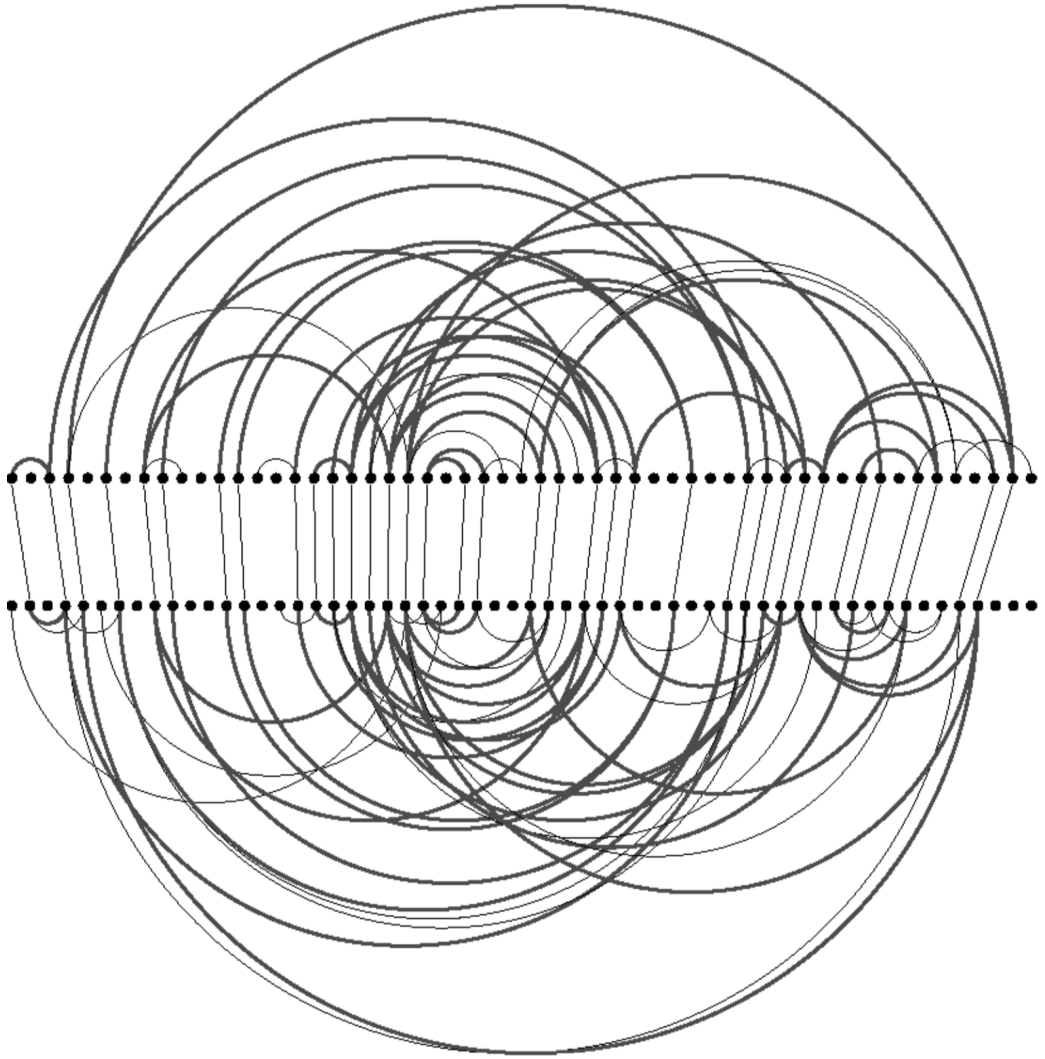
Global Alignment problem: Given strings $\mathbf{v} \in \Sigma^m$ and $\mathbf{w} \in \Sigma^n$ and scoring function δ, find alignment of $\mathbf{v}$ and $\mathbf{w}$ with maximum score

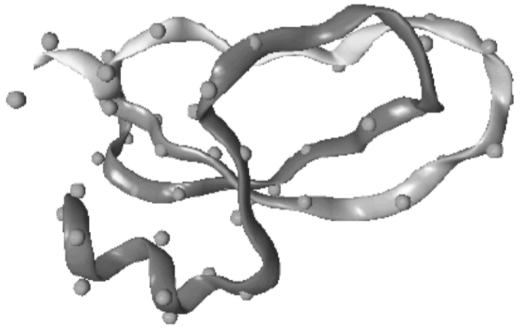
[Needleman-Wunsch algorithm]

Local Alignment problem: Given strings $\mathbf{v} \in \Sigma^m$ and $\mathbf{w} \in \Sigma^n$ and scoring function δ, find a substring of $\mathbf{v}$ and a substring of $\mathbf{w}$ whose alignment has maximum global alignment score s^* among *all* global alignments of *all* substrings of $\mathbf{v}$ and $\mathbf{w}$

[Smith-Waterman algorithm]

How to Compare Two Protein Structures?





1knt-4.0.cm: 55 residues with 43 contacts.

31 shared contacts.

1bpi-4.0.cm: 58 residues with 53 contacts.

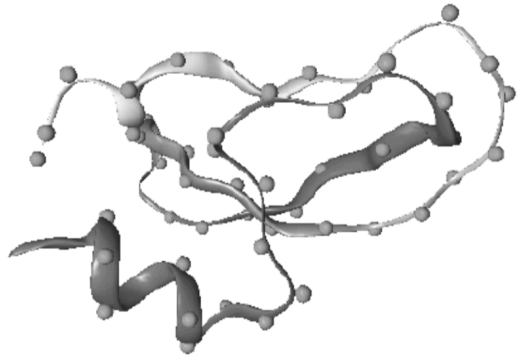


FIG. 1. An optimal alignment of two 4Å threshold contact maps of proteins `1bpi` and `1knt`.

Contact Map Overlap: Example Instance

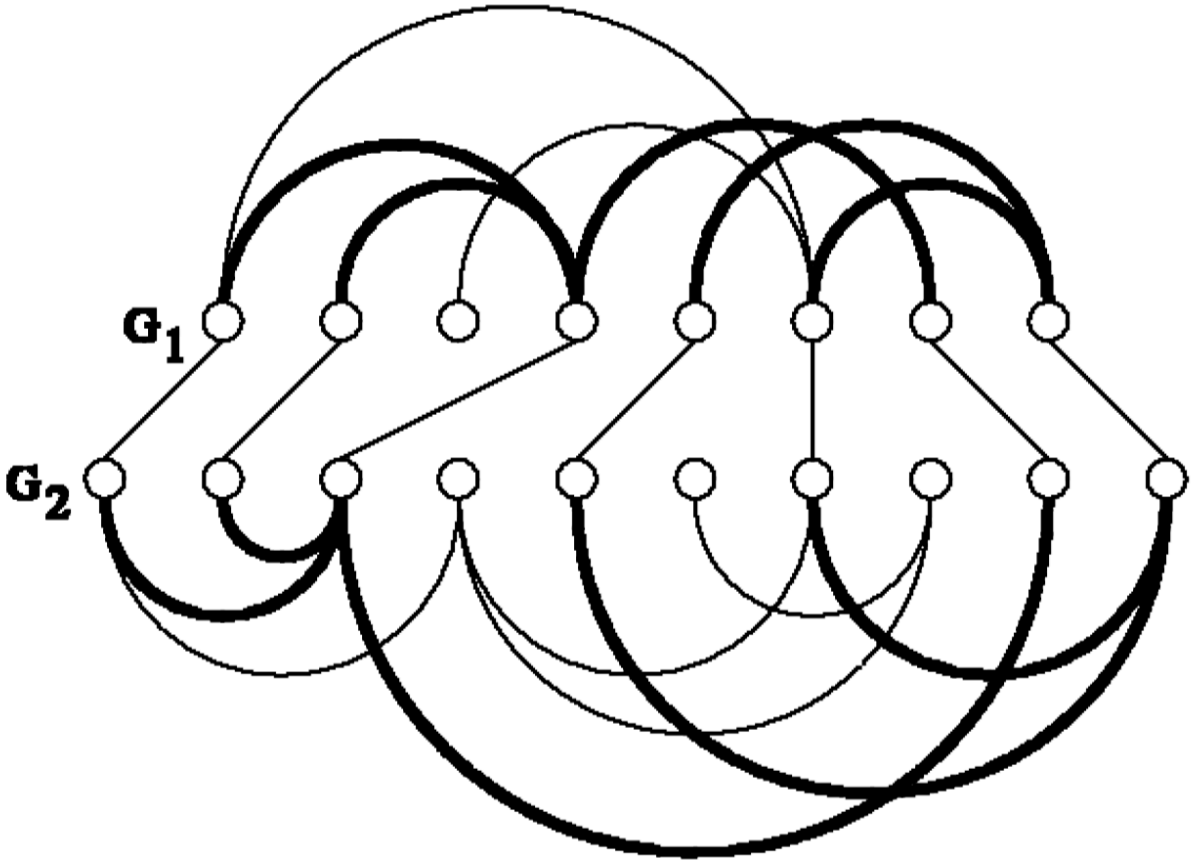


FIG. 2. An alignment of value 5.

Caprara, A., Carr, R., Istrail, S., Lancia, G., & Walenz, B. (2004). 1001 Optimal PDB Structure Alignments: Integer Programming Methods for Finding the Maximum Contact Map Overlap. *Journal of Computational Biology*, *11*(1), 27–52.

Contact Map Overlap: Equivalent Representations

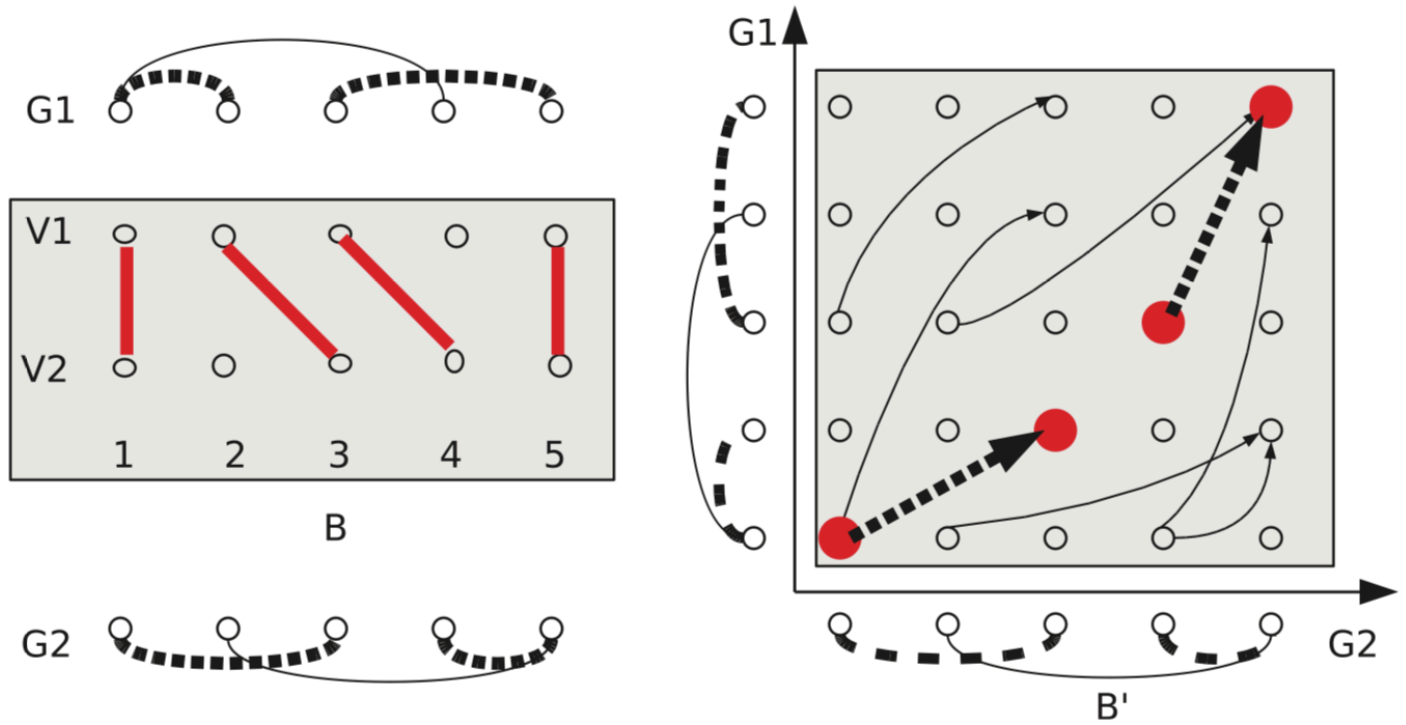


FIG. 2. Relationship between a matching in a bipartite graph B and a feasible path in the corresponding grid graph B'. **(Left)** Two contact maps $G1$ and $G2$, and a matching in the bipartite graph B (in the grey area). Note that B is a complete graph, but for the sake of simplicity only the edges of the considered matching $(M = \{(1,1)(2,3),(3,4)(5,5)\})$ are visualized. According to (1), $w(M) = 2$. **(Right)** The same matching is visualized in the grid alike graph B' as an increasing set of vertices $\{(1,1)(2,3),(3,4)(5,5)\}$ which we call a feasible path. It activates the arcs ((1,1)(2,3) and (3,4)(5,5)). The score of the path is the number of these arcs (i.e., 2 in this case).

DOI: 10.1089/cmb.2009.0196

Integer Linear Programming

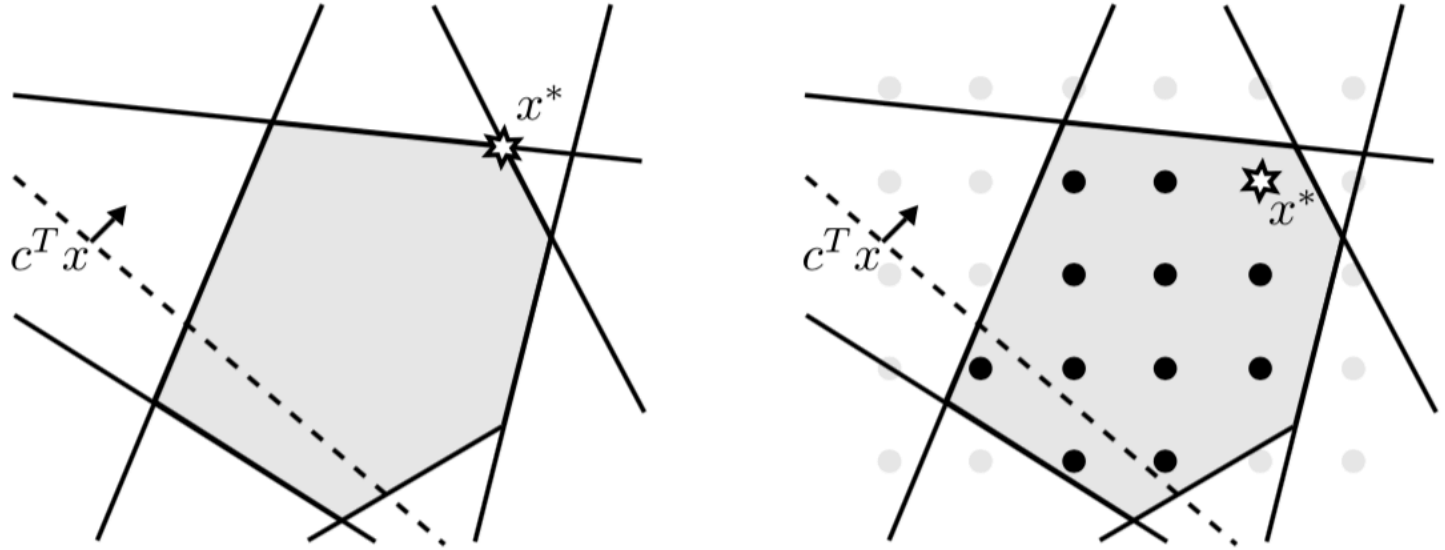


Figure 2.1: In gray, a polyhedron that is described by six constraints $a_i x \leq b_i$. The objective function $c^T x$ increases in the direction in which the arrow points. The optimal solution x^* is denoted by a star. Left: The linear program. Right: The integer linear program. Here, only the integer points within the polyhedron are feasible, which are colored black. The LP relaxation of this ILP is the LP problem that is visualized on the left side. The optimal objective function value of the LP relaxation is always an upper bound on the optimal objective function value of the ILP.

Inken Wohlers. Exact algorithms for pairwise protein structure alignment. PhD thesis, VU University Amsterdam, 2012

Separating Cutting Planes

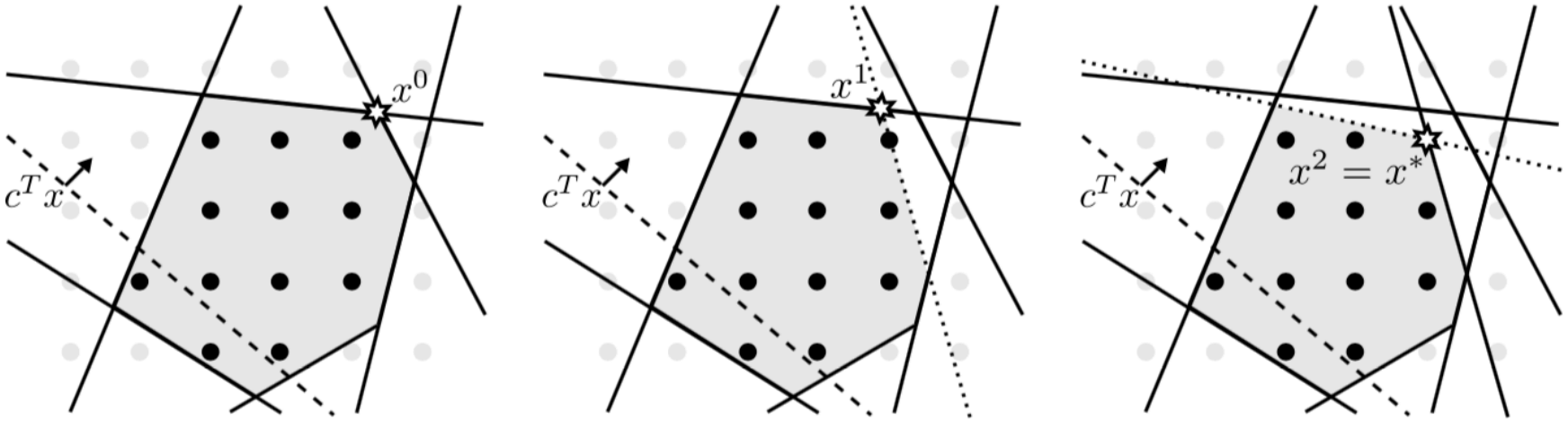


Figure 2.2: The cutting plane method solves an ILP problem. The gray area denotes the polyhedron described by the constraints of the current relaxed problem. The dashed line denotes the objective function $c^T x$ which increases in the direction in which the arrow points. The integer feasible solutions are colored black. Solving the LP relaxation, we obtain the relaxed solution x^0. After adding a cutting plane (dotted line), we obtain a new relaxed solution x^1. Finally, after adding a second cutting plane, we obtain solution x^2 which has integer value and is thus the optimal solution x^* of the ILP. A cutting plane method solves only the LP relaxation of an ILP, but here, since the optimal solution has integer value, the solution of the LP relaxation is also the solution of the ILP.

Inken Wohlers. Exact algorithms for pairwise protein structure alignment. PhD thesis, VU University Amsterdam, 2012

Cutting Plane Method

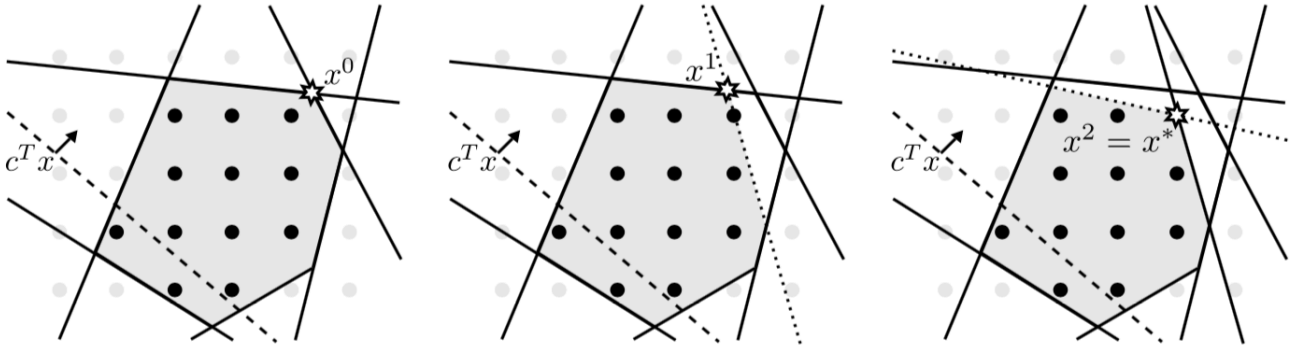


Figure 2.2: The cutting plane method solves an ILP problem. The gray area denotes the polyhedron described by the constraints of the current relaxed problem. The dashed line denotes the objective function $c^T x$ which increases in the direction in which the arrow points. The integer feasible solutions are colored black. Solving the LP relaxation, we obtain the relaxed solution x^0. After adding a cutting plane (dotted line), we obtain a new relaxed solution x^1. Finally, after adding a second cutting plane, we obtain solution x^2 which has integer value and is thus the optimal solution x^* of the ILP. A cutting plane method solves only the LP relaxation of an ILP, but here, since the optimal solution has integer value, the solution of the LP relaxation is also the solution of the ILP.

Algorithm 2 Solving an LP problem using the cutting plane method.

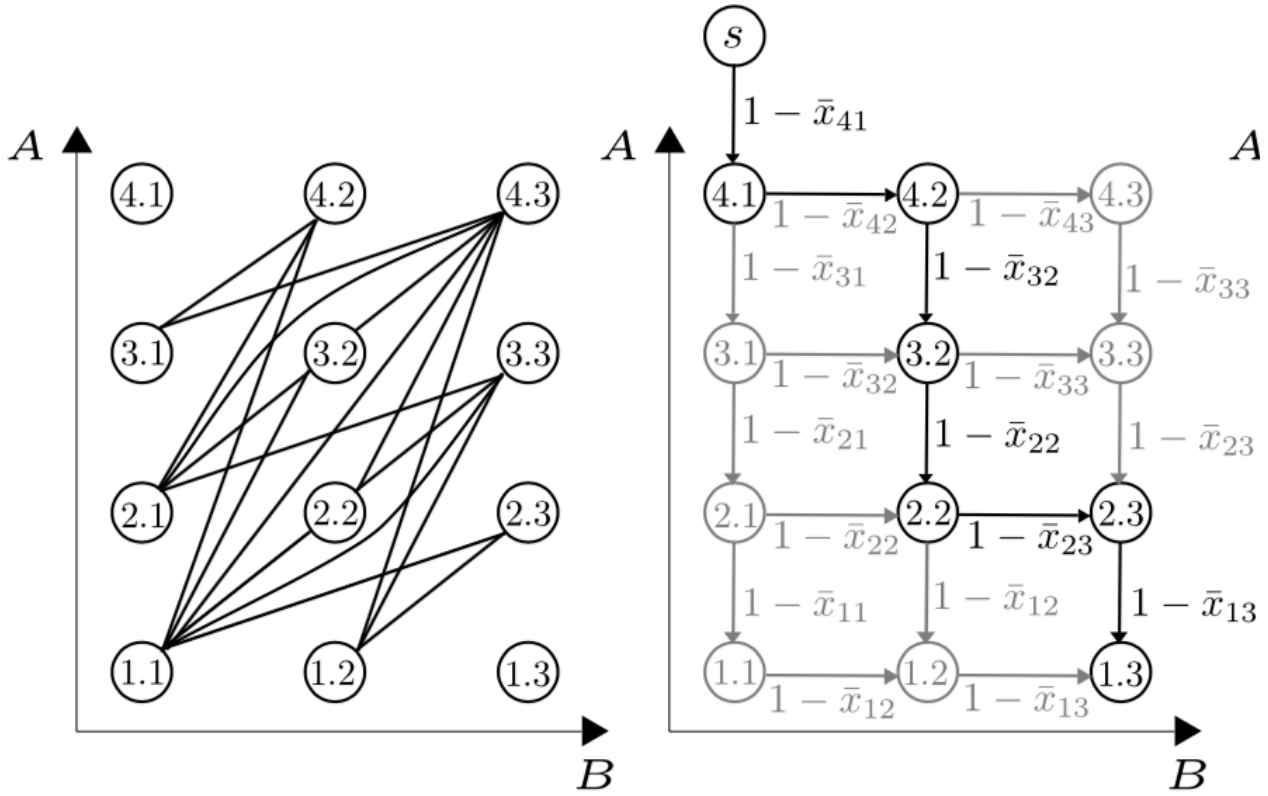
```
1: P // The original problem
2: P^t // The relaxed problem in iteration t
3: t ← 0 // Iteration
4: while True do
5:    Compute optimal solution x^t for P^t
6:    if x^t feasible for P then
7:       return x^t
8:    else
9:       Find a cutting plane a_i x ≤ b_i that all solutions of P satisfy, but not x^t
10:      P^{t+1} ← P^t with additional constraint a_i x ≤ b_i
11:      t ← t + 1
12:   end if
13: end while
```

Inken Wohlers. Exact algorithms for pairwise protein structure alignment. PhD thesis, VU University Amsterdam, 2012

Branch & Cut: Solving an ILP

- Whiteboard

Cut Separation in CMO



Inken Wohlers. Exact algorithms for pairwise protein structure alignment. PhD thesis, VU University Amsterdam, 2012

Figure 3.8: Example of the graphs in which we use shortest path computations to detect violated constraints. Left: The alignment graph $G = (V, E)$. Center: The graph $G' = (V', E')$ in which we identify a violated constraint (3.8). The shortest decreasing path is colored black, it is $C = \{4.1, 4.2, 3.2, 2.2, 2.3, 1.3\}$. If for the this path $\sum_{i.k \in C} \bar{x}_{ik} > 1$ holds, we identified a violated constraint (3.8).