Part 1

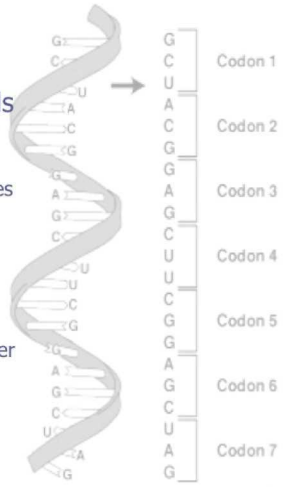
Genome and Sequences

The Genetic Code

(1) 4 Letters – ATGC

(2) A triplet code for 20 amino acids
- Four letters allow 64 different *ordered* triplets
- Several amino acids have multiple codes
- Further control codes
- Code is not universal, evolutionary distant species use different codes
- Each code has a specific translator molecule (tRNA) matching 3 bases on one side and an amino acid on the other side

(3) Start position and direction matter!



Genes

(1) Elementary coding regions on the DNA
- Sections coding particular proteins (or parts of proteins or RNA molecules)
- Some species have lots of non-coding „junk DNA"

(2) Reading process must be activated
- Specialized cells only activate a subset of the genes
- By complex regulation processes (through „docking" at the DNA, other proteins can promote or inhibit)
- DNA double-helix needs to be unfolded and unzipped

(3) A gene is a structured sequence
- Exons are sections that are decoded
- Introns are spliced out after reading and thus are ignored
- Promotors are distant regions relevant for regulation

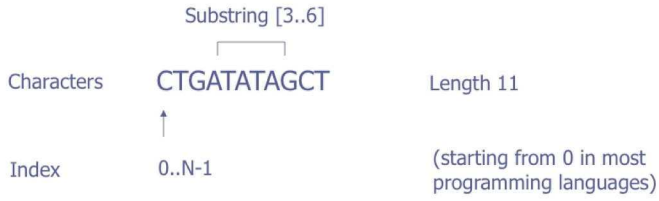
The Use of Models

(1) The genetic code is a model
- Sequence is reduced to letters
- Simplified, abstract view of the DNA molecule and its role in the organism
- Sequence of amino acids uniquely identifies a protein
- Analysis and predictions can be made on the model (-> bioinformatics)

(2) The idea of a „gene" is a model
- Genome is reduced to set of genes with functional annotation
- The idea is „something that is responsible for a set of proteins"
- There is nothing universal on the DNA saying „gene starts here"
- Actual gene performance (splicing, regulation) much more complex
- A catalog of genes/proteins helps structuring knowledge

How to find and compare genes?

String representation



The string model for genetic analysis

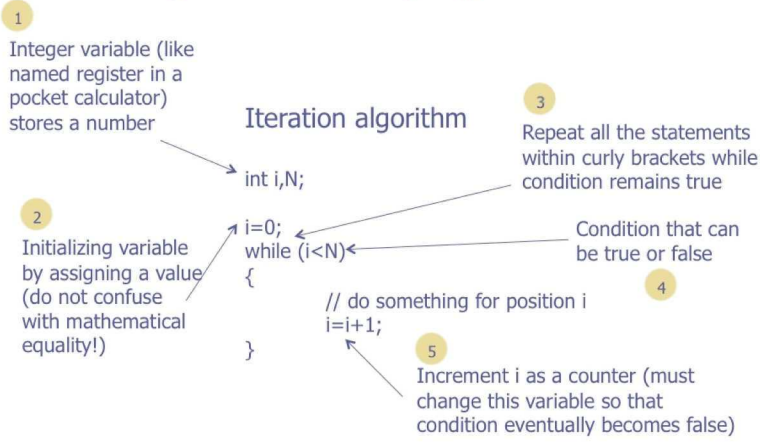
(1) What can you do with a genomic sequence string?

- Annotate a sequence with a gene (-> database)
- Collect the whole genome as a string
- Count the number/frequency of letters/bases
- Find the location of a gene in the whole sequence
- Find a degree of similarity of sequences
- Assemble overlapping sequence fragments (-> Genome Project)
- Establish differences between organisms by (dis)similarity

(2) What would you like to do with a genomic sequence string?

- Find the location of genes in a sequence
- Identify patterns that represent specific (regulatory) functions
- Identify functional sites in the encoded protein
- Predict the 3D structure of the encoded protein
- ...

Describing and executing algorithms



Language: groovy.codehaus.org

Elementary string algorithms (1)

Compare strings a and b of length N:

```
string a,b;
1 int i,N;
boolean same;

2 i=0;
while (i<N && a[i]==b[i]) 3
{
        i=i+1; 4
}
if (i>=N)
        same = true;
5 else
        same = false;
```


Programming nomenclature

string a,b;
Defining two variables of type *string*
Value could be set by
a=„TTAG"

&&
The boolean „and" operator that checks whether both conditions are true

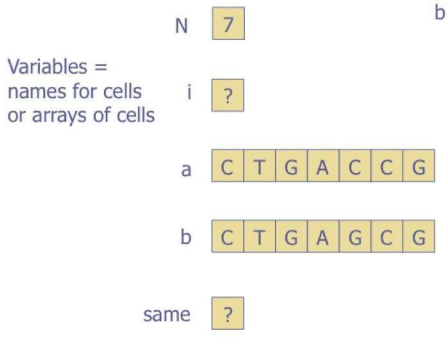
==
The mathematical „equal" operator, checking whether two numbers are the same

while (*some condition*) { *a group of statements to be repeated* }
The general loop statement. While condition is true, statements are executed over and over

if (*some condition*) *statement1* else *statement2*
A conditional statement that, depending on the condition, executes either one or the other

Computer memory

N = 7;
a = "CTGACCG";
b = "CTGAGCG";



Each memory cell contains a number or character code, which is initially undefined

Elementary string algorithms (2)

Search string b (length M) in a (length N):

```
int i, N, M;
boolean found = false;
int offset = 0;     // first try for the start pos of b in a

while  (N-M >= offset && found == false)
{
        i=0;
        while (i<M && a[i+offset]==b[i])
        {
          i=i+1;
        }
        if (i>=M)
                found = true;
        else
                offset=offset+1;
}
```

Algorithmic complexity

(1) *O* (F(n)) means runtime on large n is always less than cF(n)
- There may be a constant overhead that is relevant for small n
- c is an arbitrary constant

(2) String comparison is *O* (N)
- Linear time, c is roughly the time needed for one loop

(3) Simple string search is *O* (MN)
- For large N and small M, may need to try all N positions and compare with M characters
- On the average, only half of N needs to be checked, so c could be half the time for one inner loop

(4) Polynomial time complexity is still „efficient"
- Exponential time is no longer computationally tractable

More efficient string search

(1) Boyer-Moore skips some intermediate positions
- Compares from right to left
- If there is a mismatch, it is a property of the pattern how many positions it can advance for the next comparison
- CT**GAT**ATAGCT
- TA**GAT**
- We can advance the pattern by 4 positions as GAT does not appear in the pattern again (always if there is a mismatch at position 2)

(2) Usually trade time for space
- Preprocessing the pattern (useful if pattern is much smaller than the search string)
- Array of numbers for each position in the pattern

1.1

Sequence alignment

Sequence alignment

(1) Biological sequences (DNA, protein) may only be similar instead of equal
- Sequencing errors
- Mutations (changes, insertions, deletions)
- Comparison across evolutionary changes
- Comparison with functionally related sequences

(2) How to describe similarity?
- Amino acids might be replaced by chemically similar ones
- Long gaps might not be relevant (spliced away, hidden in a functionally inactive area of the protein)

Pairwise alignment

GFTGATATAGFT
GGGTGATTAGFT

only 6 matches

_GFTGATATAGFT
GGGTGAT_TAGFT

10 matches when gaps are allowed

__GFTGATATAGFT
GGG_TGAT_TAGFT

Same number of matches, more gaps

Could be more probable if the F-by-G substition disturbes function more than a gap

Examples for pairwise alignment

human alpha globin vs. human beta globin

human alpha globin vs. leghaemoglobin from yellow lupin (evolutionary and functionally related)

human alpha globin vs. F11G11.2 (unrelated)

Scoring

(1) Numerical function to decide which alignment is the most relevant
- Higher scores = higher probability

(2) Scoring function is a model of the biophysical situation
- Which amino acids are similar with respect to molecular forces?
- Which amino acids could be replaced without changing the protein structure?
- Which amino acids are less relevant for the overall structure (i.e. which are not involved in an active binding domain)?

(3) Scores can be derived from biological observations
- Depending on the frequency of pairs of characters occurring in alignments that are biologically „approved" as relevant
- Sort of „machine learning"

Scoring scheme

(1) Score
- $\sum s(a_i,b_i)$ where s(a,b) is the substitution matrix entry for the amino acids a and b
- The substitution matrix contains the degree of similarity of the two amino acids with respect to their effect on the protein function

(2) Gap penalty
- Fixed value for each gap position (additive)
- Penalizes long gaps
- Better approximation: fixed value for first gap character and smaller addition for every further one

(3) Simple algorithm
- Enumerate all possible alignments and calculate the score

Substitution matrices

(1) Matrix contains likelihood p of each aligned pair compared to random appearance (frequency q)
- log $(p_{ab}/q_a q_b)$ scaled and rounded to the nearest integer

BLOSUM50 matrix

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	5																			
R	-2	7																		
N	-1	-1	7																	
D	-2	-2	2	8																
C	-1	-4	-2	-4	13															
Q	-1	1	0	0	-3	7														
E	-1	0	0	2	-3	2	6													
G	0	-3	0	-1	-3	-2	-3	8												
H	-2	0	1	-1	-3	1	0	-2	10											
I	-1	-4	-3	-4	-2	-3	-4	-4	-4	5										
L	-2	-3	-4	-4	-2	-2	-3	-2	-3	2	5									
K	-1	3	0	-1	-3	2	1	-2	0	-3	-3	6								
M	-1	-2	-2	-4	-2	0	-2	-3	-1	2	3	-2	7							
F	-3	-3	-4	-5	-2	-4	-3	-4	-1	0	1	-4	0	8						
P	-1	-3	-2	-1	-4	-1	-1	-2	-2	-3	-4	-1	-3	-4	10					
S	1	-1	1	0	-1	0	-1	0	-1	-3	-3	0	-2	-3	-1	5				
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	2	5			
W	-3	-3	-4	-5	-5	-1	-3	-3	-3	-3	-2	-3	-1	1	-4	-4	-3	15		
Y	-2	-1	-2	-3	-3	-1	-2	-3	2	-1	-1	-2	0	4	-3	-2	-2	2	8	
V	0	-3	-3	-4	-1	-3	-3	-4	-4	4	1	-3	1	-1	-3	-2	0	-3	-1	5

The problem of optimum alignment

(1) The scores are only a crude approximation
- We do not know the internal forces and mechanisms
- Co-evolution is also an effect when scoring evolutionary conservation
- We treat amino acids independently

(2) The number of possible alignments is exponential with the length of the sequence!
- It is not possible to just go through all the possible alignments, score them, and keep the best one.

(3) How do we find out where to put gaps
- The solution is a special algorithm that can do this provided the score is additive
- Additive score means that the scores for all pairs of amino acids or amino acids with gaps are added up
- An additive score is not able to model neighbour effects, such as „A" is less probable if the neighbour is „T".

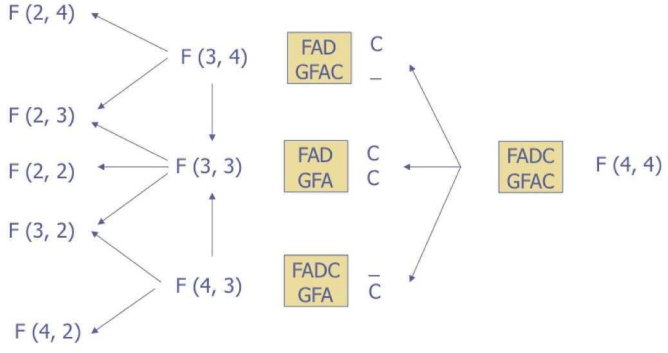
Dynamic programming

(1) Best alignment can be calculated based on the best alignments of prefixes of the two strings
- Uses additive property of score
- Let F (i, j) be the score of the best alignment between a[0..i-1] and b[0..j-1]

(2) Three possible cases
- s: substitution matrix, d: gap penalty
- a[i-1] and b[j-1] match:
  F (i,j) = F (i-1, j-1) + s (a[i-1], b[j-1])
- a[i-1] is aligned to a gap:
  F (i,j) = F (i-1, j) - d
- b[j-1] is aligned to a gap:
  F (i,j) = F (i, j-1) - d

(3) Iteratively calculate F as the maximum of the three cases

Example



Needleman-Wunsch algorithm

```
for (i in 0..M) F [i][0] = -i * d;
for (j in 0..N) F [0][j] = -j * d;
for (i in 1..M)
        for (j in 1..N)
                F [i][j] = max (
                        F [i-1][j-1] + s (a [i-1], b[j-1]),
                        F [i-1][ j ] - d,
                        F [i]  [j-1] - d);
```

a = HEAGAWGHEE
b = PAWHEAE

		H	E	A	G	A	W	G	H	E	E
	0	-8	-16	-24	-32	-40	-48	-56	-64	-72	-80
P	-8	-2	-9	-17	-25	-33	-41	-49	-57	-65	-73
A	-16	-10	-3	-4	-12	-20	-28	-36	-44	-52	-60
W	-24	-18	-11	-6	-7	-15	-5	-13	-21	-29	-37
H	-32	-14	-18	-13	-8	-9	-13	-7	-3	-11	-19
E	-40	-22	-8	-16	-16	-9	-12	-15	-7	3	-5
A	-48	-30	-16	-3	-11	-11	-12	-12	-15	-5	2
E	-56	-38	-24	-11	-6	-12	-14	-15	-12	-9	1

Example: global alignment

F(4,1):

F(4,0) = - 32 d = - 8
F(3,1) = - 17 d = - 8
F(3,0) = - 24 s(P,G) = - 2;

F(4,1) = max(- 40,- 25,- 26) = - 25

F(6,3):

F(5,3) = - 15 d = - 8
F(6,2) = - 28 d = - 8
F(5,2) = - 20 s(W,W) = 15

F(6,3) = max(- 5,- 36,- 23) = - 5

Optimal global alignment:

HEAGAWGHE -E
- - P -AW- HEAE

		H	E	A	G	A	W	G	H	E	E
	0	-8	-16	-24	-32	-40	-48	-56	-64	-72	-80
P	-8	-2	-9	-17	-25	-33	41	-49	-57	-65	-73
A	-16	-10	-3	-4	-12	-20	-28	-36	-44	-52	-60
W	-24	-18	-11	-6	-7	-15	-5	-13	-21	-29	-37
H	-32	-14	-18	-13	-8	-9	-13	-7	-3	-11	-19
E	-40	-22	-8	-16	-16	-9	-12	-15	-7	3	-5
A	-48	-30	-16	-3	-11	-11	-12	-12	-15	-5	2
E	-56	-38	-24	-11	-6	-12	-14	-15	-12	-9	1

Recovering the alignment

(1) Backtracking from F(M, N)

- Going back to the cell from which the maximum was calculated
- Depending on the three cases, add either a gap on one side or a matched pair
- Either use recursion or build backwards
- Exercise: write down the algorithm

(2) Multiple solutions possible

- Whenever maximum of the three choices is not unique

Overlap matches

(1) No penalty for gaps at the beginning or end

- When only overlaps are assumed
- For example, when assembling DNA fragments (shotgun sequencing)

(2) Small algorithmic differences

- Initialization of top and left border to 0 (no penalty)
- Take the maximum from the right and bottom border (instead of the bottom-right corner) to find the best partial match
- Exercise: write down the modified algorithm

Local alignment (Smith-Waterman)

```
for (i in 0..M) F [i][0] = 0;
for (j in 0..N) F [0][j] = 0;
for (i = 1; i <= length (a); i++)
        for (j = 1; j <= length (b); j++)
                F (i, j) = max (0,
                        F (i-1, j-1) + s (a[i-1], b[j-1]),
                        F (i-1, j) – d),
                        F (i, j-1) – d));
```

Alignment ends in maximum F

Regard only local matches with positive scores (Random s(a,b) must be negative, at least one must be positive)

Optimal local alignment: AWGHE / AW -HE

		H	E	A	G	A	W	G	H	E	E
	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	5	0	5	0	0	0	0	0
W	0	0	0	0	2	0	20	12	4	0	0
H	0	10	2	0	0	0	12	18	22	14	6
E	0	2	16	8	0	0	4	10	18	28	20
A	0	0	8	21	13	5	0	4	10	20	27
E	0	0	6	13	18	12	4	0	4	16	26

Repeated local alignment

```
F (0, j) = 0;
for (i = 1; i <= length (a); i++)
        F (i, 0) = max (F (i-1, 0), F (i-1, j) – T);
        for (j = 1; j <= length (b); j++)
                F (i, j) = max (F(i, 0),
                        F (i-1, j-1) + s (a[i], b[j]),
                        F (i-1, j) – d),
                        F (i, j-1) – d));
```

Positive threshold T (= 20)

Asymmetric

Going back from F(n+1,0)

HEAGAWGHEE
HEA AW -HE

		H	E	A	G	A	W	G	H	E	E
	0	0	0	0	1	1	1	1	1	3	9
P	0	0	0	0	1	1	1	1	1	3	9
A	0	0	0	5	1	6	1	1	1	3	9
W	0	0	0	0	2	1	21	13	5	3	9
H	0	10	2	0	1	1	13	19	23	15	9
E	0	2	16	8	1	1	5	11	19	29	21
A	0	0	8	21	13	6	1	5	11	21	28
E	0	0	6	13	18	12	4	1	5	17	27

Affine gap scores

(1) Lower penalty for longer gaps

- d = initial gap penalty, e = penalty for each extension

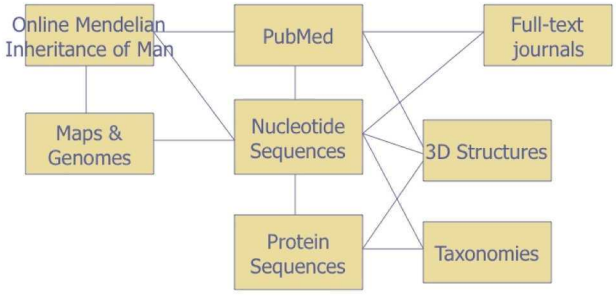
(2) Dynamic programming algorithm with $O(n^2)$ possible

- M (i, j): best score given that a[i] and b[j] match
- Ia (i, j): best score given that a[i] is matched to a gap
- Ib (i, j): best score given that b[j] is matched to a gap
- M (i, j) = max (M (i-1, j-1) + s (a[i], b[j]),
  Ia (i-1, j-1) + s (a[i], b[j]),
  Ib (i-1, j-1) + s (a[i], b[j]))
- Ia (i, j) = max (M (i-1, j) – d,
  Ia (i-1, j) – e);
- Ib (i, j) = max (M (i, j-1) – d,
  Ib (i, j-1) – e);

1.2

Databases and heuristic algorithms

Databases at the NCBI



National Center for Biotechnology

Sequence databases

(1) Store any sequences and fragments that have been found
(2) Unique accession key (arbitrary number)
(3) Attributes such as source, species, etc.
(4) One attribute is the sequence string

Example GenBank entry (1)

```
XX
AC   X04751;
XX
SV   X04751.1
XX
DT   07-JUN-1987 (Rel. 12, Created)
DT   10-FEB-1999 (Rel. 58, Last updated, Version 5)
XX
DE   Rabbit alpha-1-globin gene to theta-1-globin pseudogene region
XX
KW   alpha-1-globin; alpha-globin; globin; pseudogene; repetitive sequence;
KW   tandem repeat; theta-1-globin; theta-globin.
XX
OS   Oryctolagus cuniculus (rabbit)
OC   Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi; Mammalia;
OC   Eutheria; Lagomorpha; Leporidae; Oryctolagus.
XX
RN   [1]
RP   1-4028
RA   Hardison R.C.;
RT   ;
RL   Submitted (02-FEB-1987) to the EMBL/GenBank/DDBJ databases.
RL   Hardison R.C., Pennsylvania State University, Althouse Laboratory,
RL   University Park, Pennsylvania 16802, USA.
```


Example GenBank entry (2)

```
XX
RN   [2]
RP   1-4028
RX   MEDLINE; 86085923.
RA   Cheng J.-F.F., Raid L., Hardison R.C.;
RT   "Isolation and nucleotide sequence of the rabbit globin gene cluster
RT   psi-zeta-alpha-1-psi-alpha: Absence of a pair of alpha-globin genes
RT   evolving in concert";
RL   J. Biol. Chem. 261:839-848(1986).
XX
DR   EPD; EP11096; OC_HBA.
DR   SWISS-PROT; P01948; HBA_RABIT.
XX
CC   Submitted data [2] include some corrections to published seq. [1].
CC   Referring to the authors the sequence from pos. 50 to 70 may not
CC   be completely accurate due to reading problems of the sequencing
CC   gels.
CC   Theta-1 pseudogene was formerly called psi alpha.
CC   Data kindly reviewed (15-Jun-1987) by Hardison R.C.
XX
```


Example GenBank entry (3)

```
FH   Key             Location/Qualifiers
FH
FT   source          1..4028
FT                   /db_xref="taxon:9986"
FT                   /organism="Oryctolagus cuniculus"
FT   precursor_RNA   150..861
FT                   /note="primary transcript od alpha-1-globin"
FT   exon            150..280
FT                   /number=1
FT   CDS             join(186..280,358..562,646..774)
FT                   /db_xref="SWISS-PROT:P01948"
FT                   /product="alpha-1-globin"
FT                   /protein_id="CAA28447.1"
FT                   /translation="MVLSPADKTNIKTAWEKIGSHGGEYGAEAVERMFLGFPTTKTYFP
FT                   HFDFTHGSEQIKAHGKKVSEALTKAVGHLDDLPGALSTLSDLHAHKLRVDPVNFKLLSH
FT                   CLLVTLANHHPSEFTPAVHASLDKFLANVSTVLTSKYR"
```

Example GenBank entry (4)

```
FT   intron          281..357
FT                   /number=1
FT   exon            358..562
FT                   /number=2
FT   intron          563..645
FT                   /number=2
FT   exon            646..861
FT                   /number=3
FT   polyA_signal    841..846
FT   polyA_site      861..861
FT   repeat_region   1542..1675
FT                   /note="region of 5 x 25bp tandem repeat 1"
FT   repeat_region   3067..3133
FT                   /note="region of 7 tandem repeat 2 (9-10bp)"
FT   CDS             3139..3744
FT                   /pseudo
FT                   /product="theta-1-globin"
FT   polyA_signal    3803..3808
FT   polyA_site      3818..3818
FT                   /note="put. polyA site (found by homology to alpha-1)"
XX
```


Example GenBank entry (5)

```
SQ   Sequence 4028 BP; 685 A; 1359 C; 1310 G; 674 T; 0 other;
     gcggggccgg gtcccaggca gacgccgcga gggcgccccc agcggtggcg gccgccgccg        60
     cgccccgccg cgccggccaa tgagcggggc cccgctgggc gtgcccgcag cacctcgggc       120
     ttaaaagcgc cgcgcagtct gggctccgca cacttctggt ccagtccgac tgagaaggaa       180
     ccaccatggt gctgtctccc gctgacaaga ccaacatcaa gactgcctgg gaaaagatcg       240
     gcagccacgg tggcgagtat ggcgccgagg ccgtggagag gtgaggaccc ccgccccgcc       300
     ccgccccgcc cgagcccgcc ggcgccgcgc cccgctcacg gcctcctgtc cccgcaggat       360
     gttcttgggc ttccccacca ccaagaccta cttcccccac ttcgacttca cccacggctc       420
     tgagcagatc aaagcccacg gcaagaaggt gtccgaagcc ctgaccaagg ccgtgggcca       480
     cctggacgac ctgcccggcg ccctgtctac tctcagcgac ctgcacgcgc acaagctgcg       540
     ggtggacccg gtgaatttca aggtgagccc gcagcccggc tgggagcgtc gcgggggtcg       600
     gcggtccccg accacaccca ccgacgtccg cccctctctc tgcagctcct gtcccactgc       660
     ctgctggtga ccctggccaa ccaccacccc agtgaattca cccctgcggt gcacgcctcc       720
     ctggacaagt tcctggccaa cgtgagcacc gtgctgacct ccaaatatcg ttaagctgga       780
     gcctgggagc cggcctggcc ctccgccccc cccacccccg cagcccaccc ctggtctttg       840
     aataaagtct gagtgagtgg ccgacagtgc ccgtggagtt ctcgtgacct gaggtgcagg       900
     gccggcctag ggacacgtcc gtgcacgtgc cgaggccccc tgtgcagctg caagggacag       960
     gagtgggcaa ccggctggtt ccttccttcc tgcttgcaag tccacgaggg gctgctgaaa      1020
     gaaccccca cacacacatg cacacactcg tgccactcgg ctgcctccag cctgggtccc      1080
     ....
```

Finding related sequences

(1) Find related sequences to a target sequence
- Process all entries of the database and do matching
- Computationally expensive

(2) Faster alignment algorithms needed
- Optimality cannot be guaranteed
- Focus on matching ungapped segments
- FAST and BLAST

BLAST

(1) Segment pairs
- Segment pair: Two aligned subsequences without gaps
- Find all high-scoring segment pairs between two sequences
- Similar to a gapped sequence without scoring gaps
- Heuristics focus on locally conserved/related sequences

(2) Steps of the algorithm
- Find all words (e.g. length = 4 characters) that match somewhere in the query sequence with a score > T
- Find occurrences of these words (seeds) in the comparison string
- Extend seeds in both directions until score drops below a fraction of the maximum so far
- Report all segment pairs with score > S

Further parameter considerations

(1) Low-complexity regions
- Regions of low variation (only few different amino acids with high repetition rate)
- Can produce high scoring hits
- Biologically „assumed" to be irrelevant/non-functional
- Can be filtered (substituted by X) before query

(2) Different substitution matrices
- Probabilities calculated for specific number of evolutionary steps
- PAM60 means 60 changes of the sequence (vs. PAM120)
- Higher numbers for dissimilar sequences, lower numbers for related sequences

FAST

(1) Find high scoring offset
- E.g., s = H A R F Y A A Q I V L

(2) Lookup tuples (ktup = 1 or 2) and record offset
- A (2, 6, 7), F (4), H (1), I (9), L (11), Q (8), R (3), V (10), Y (5)

(3) Scan database string and count offsets
- E.g., t = V D M A A Q I A
- Pos 1 (V): 10 – 1 = 9: [9]
- Pos 4 (A): 2 – 4 = -2, 6 – 4 = 2, 7 – 4 = 3: [-2, 2, 3]
- Pos 5 (A): [-3, 1, 2], Pos 6 (Q): [2], Pos 7 (I): [2], Pos 8 (A): [-6, -2, -1]
- Offset 2 occurs 4 times

(4) Called „diagonal method"
- Find diagonal in the dynamic programming matrix

FAST (2)

(1) Heuristically join tuples into regions (gapless alignments)
- E.g., A A Q I with offset 2

(2) Rescore regions with substitution matrix
- Best = initial score

(3) Finally recalculate using dynamic programming restricted to a band around the diagonal found

1.3

Recognizing signals

Recognizing signals in the sequence

(1) Signals are functional sites
- E.g., start/stop, intron to extron (splice site), etc.

(2) Sometimes signals follow characteristic patterns
- E.g. splice site A G G T (A | G) A G T

(3) Describing patterns
- Regular expressions
- Statistical models

(4) Recognizing patterns

(5) Learning patterns from examples
- Determining statistical coefficients, e.g. transition probabilities

Regular expressions

(1) Describe a set of words (language)

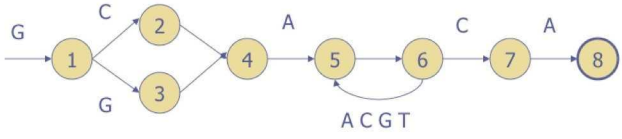
- The letters A C G T denote a word consisting of that letter
- Concatenation of two expressions x and y denotes all concatenations that can be built from words denoted by x and y
- (x | y) denotes all words denoted either by x or y
- { x } * denotes 0 or more repetitions of the words denoted by x

(2) Example: G (C | G) A { G | A | C | T }* C A

- Denotes all DNA sequences that start with either G C A or G G A and end with C A

Finite automata

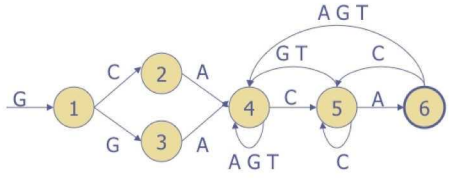
(1) Efficient match for words



Non-deterministic automaton

Word is accepted if there exists a corresponding path through the graph from start to end state

Deterministic finite automata



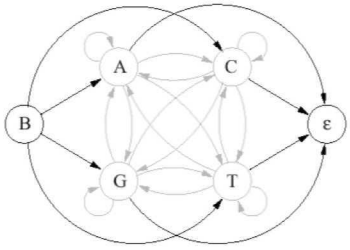
Can always be constructed, efficient table representation

	0	1	2	3	4	5	6
A	x	x	4	4	4	6	4
C	x	2	x	x	5	5	5
G	1	3	x	x	4	4	4
T	x	x	x	x	4	4	4

Markov models

(1) Going from a fixed language to transition probabilities

- A Markov chain is a process where the probability of appearance of a state (character) depends only on the previous state (character), not on the complete history



Define transition probabilities for each state transition (outgoing probabilities must sum to 1)

Example: CpG islands

(1) The probability of C G sequences in the genome is lower than random

- Reason: C in this combination is typically methylated and has a tendency to mutate into T
- Methylation is suppressed in biologically interesting regions, such as around promotors and start regions of genes. The probability of C G sequences is higher there (CpG islands)

(2) A Markov chain can distinguish between CpG islands and regular sequences

- Take a number of example sequences of each and calculate the transition probabilities (relative frequency vs. other pairs)

Example transition probabilities

+	A	C	G	T
A	0.180	0.274	0.426	0.120
C	0.171	0.368	0.274	0.188
G	0.161	0.339	0.375	0.125
T	0.079	0.355	0.384	0.182

CpG islands

-	A	C	G	T
A	0.300	0.205	0.285	0.210
C	0.322	0.298	0.078	0.302
G	0.248	0.246	0.298	0.208
T	0.177	0.239	0.292	0.292

Regular sequences

Log-likelihood ratio for a transition: logarithm of quotient between +model and –model (e.g., log (0.274/0.078) for an observed C G transition)

To score a region, sum up the log-likelihood ratios of the occurring transitions and divide by the length, positive results are indicators for the +model

Sequence families

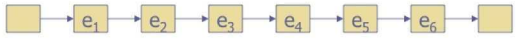
(1) Multiple alignment of related sequences
- E.g., multiple proteins with a known similar structure
- Manually align along structural information (loops and helices)
- Manually align key positions with known functionality

(2) Hidden Markov models to describe the „pattern"
- To check whether all of the structural elements are conserved
- Thus including „biological semantics" and not only substitution probabilities

(3) Profile HMMs
- Given manual alignment of example sequences
- Build model of structural features
- Estimate model parameters from example sequences
- Calculate most probable path and probability for new sequences

Position-specific score matrices

(1) Simple model for the position-specific probabilities of short ungapped segments
- e_i (a): probability that the amino acid a is observed in position i
- Equivalent to a HMM with n states



(2) Can be used to find pattern by scoring a segment from a larger sequence
- Iterate to find high-scoring segments
- Known segments can be stored in a database (BLOCKS)

Summary: Recognizing signals

(1) Regular grammars
- For short patterns
- Deterministic

(2) Position-specific scoring matrices
- Also called blocks or matrices
- For ungapped longer blocks

(3) Profile HMMs
- For carefully annotated patterns
- Most powerful, but require careful parameter estimation

1.4

Phylogenetic trees

Phylogenetic trees

(1) Phylogeny = relationship between species
- Phylogenetic tree: derivation of evolutionary relationship
- Genome sequences can be used to estimate phylogeny

(2) Genetic phylogeny not coincident with species phylogeny
- Because of events like gene duplication
- Orthologues = genes diverged through speciation
- Paralogues = genes diverged through e.g., gene duplication

(3) Tree has nodes and edges
- Edges have a distance that indicates the amount of change between species/sequences
- Edge length does not necessarily correspond exactly to evolutionary time periods (different change rates)

UPGMA clustering

(1) Computes binary tree from set of leafs and distances
- Building pairs of nearest nodes or node clusters
- Assume that distance to all leaves is the same (constant molecular clock)

(2) Initialization
- Assign each sequence i to its cluster C_i
- Define a leaf at height 0 for each sequence i

(3) Iteration while there is more than one cluster
- Find two clusters with minimal distance (average between all possible pairs)
- Join the clusters as C_k and calculate its distance to all others
- Place the new node k at the the height of half the cluster distance

Distance measures

(1) Based on alignment of sequences
- Fraction f of positions that differ
- A random alignment gives about f=0.75

(2) More realistic estimate
- Jukes-Cantor distance = $-0.75 \log (1 - 4f/3)$
- Approaches infinity as f goes towards 0.75

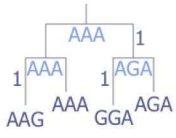
(3) UPGMA assumes additive distance
- Distance between any leaves is sum of paths conneting them
- Automatically constructed by the algorithm

Parsimony

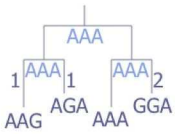
(1) Build a tree that minimizes the number of substitutions

- Enumerate all possible trees (exponential)
- Generate trees heuristically (until good enough)

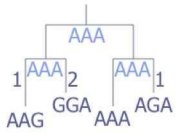
Example:
AAG
AAA
GGA
AGA



cost = 3



cost = 4



Estimating cost of tree

(1) Traditional parsimony

- Cost of one replaced letter is 1

(2) Walk recursively down the tree

- Keep minimal costs C and list of minimal-cost residues R_k at each node
- Start with the root node k = 2n – 1 and C = 0

(3) Recursion for R_k and C

- If k is a leaf:
  R_k = {assigned sequence at k}
- Otherwise:
  Compute R_i and R_j for the daughter nodes i and j
  If $R_i \cap R_j$ is empty:
  $R_k = R_i \cup R_j$;
  increment C;
  Else:
  $R_k = R_i \cap R_j$

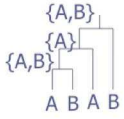
Traceback procedure

(1) To assign possible residues to each node

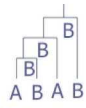
- Pick one of the minimal-cost root assignments
- Go down the tree
- Pick either the same assignment for the daughter nodes if possible
- Otherwise pick any of the minimal-cost assignments of this node

(2) Not all possible assignments can be recovered

- An additional cost down the tree can be recovered higher up
- Can be solved by keeping a list of residues at each node that have a cost of 1 more than the minimum



Recoverable

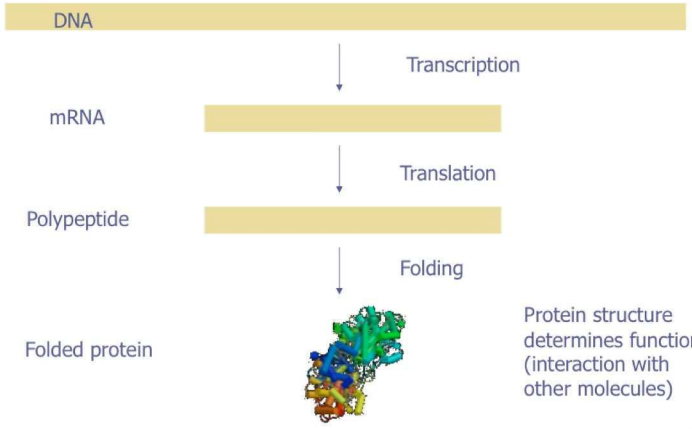


Non-recoverable, same cost

Part 2

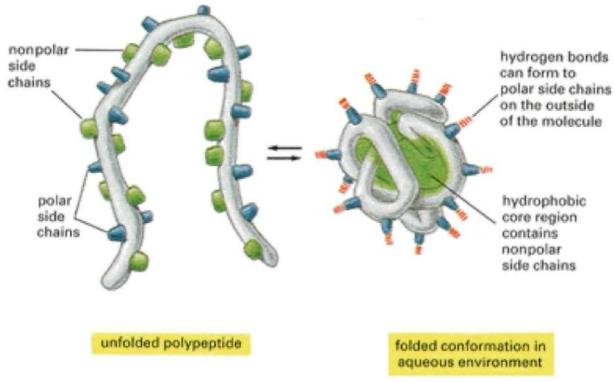
Structures and Proteins

Gene to proteins

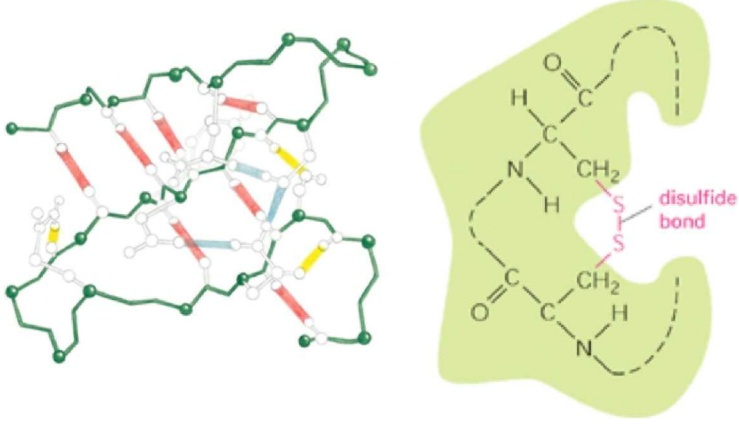


Protein structure determines function (interaction with other molecules)

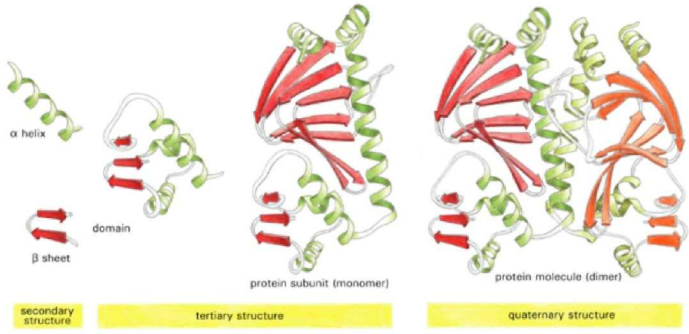
Protein folding



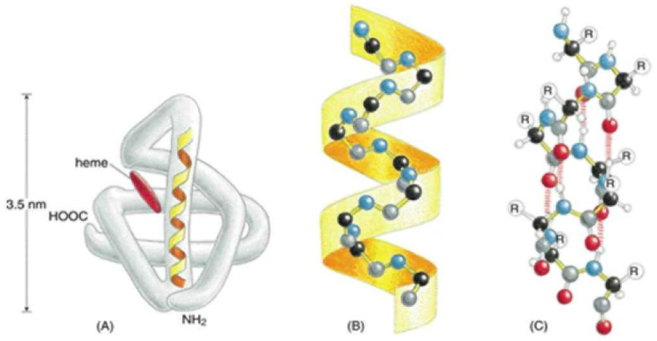
Hydrogen bonds/Disulfide bonds



Protein structure levels



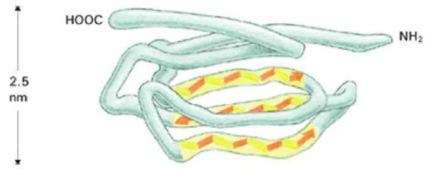
Alpha helices

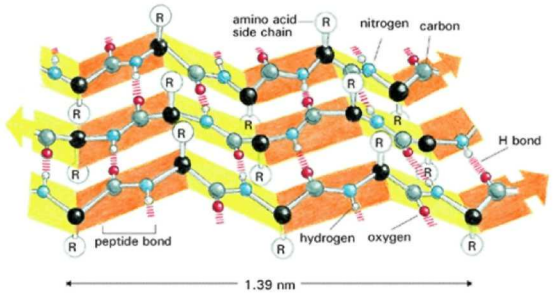


Energetically favourable configuration of the bond between amino acids

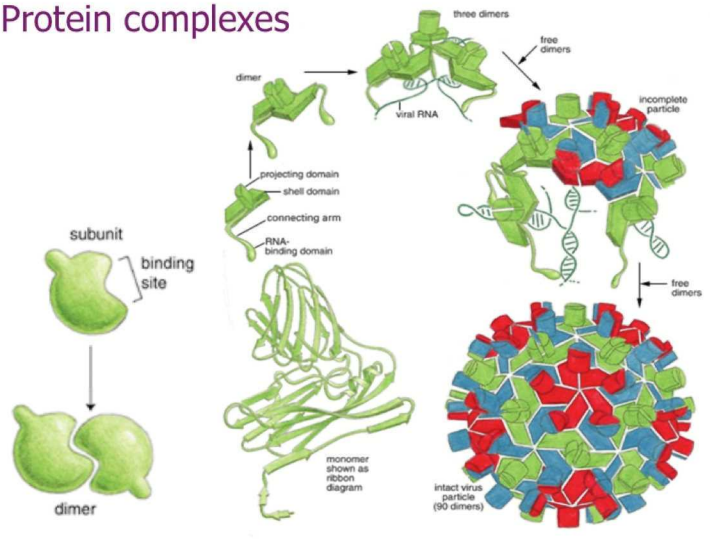
Beta sheets

Further favourable configuration



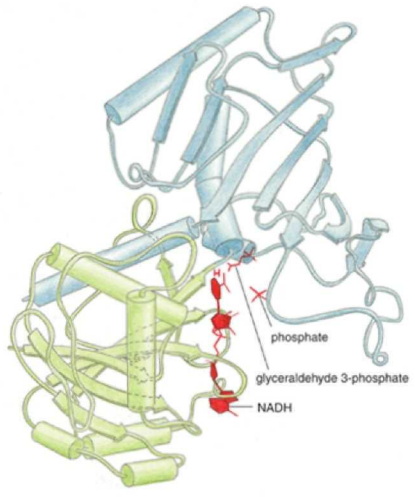


Protein complexes



Ligand binding between two domains

Catalytic function



2.1

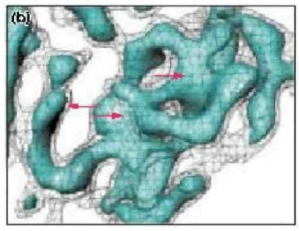
Protein structure determination

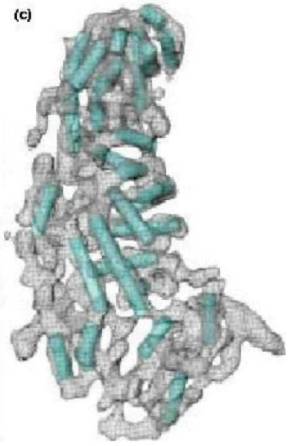
Transmission electron microscopy

1. Measuring electron diffraction on cryofixated molecules

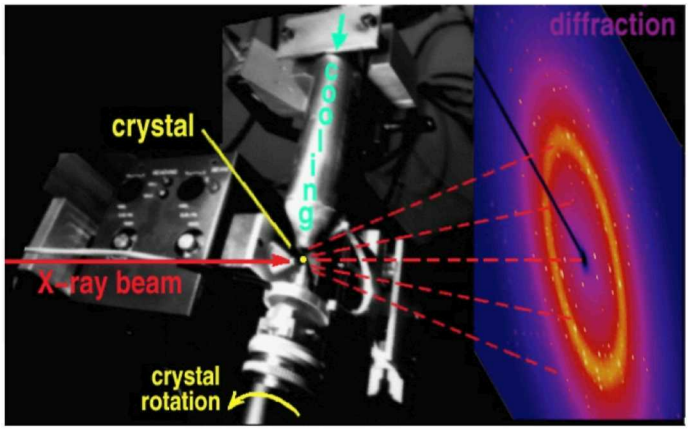
2. Reconstructing electron density around atoms

3. Visualizing surfaces of equal density (isosurfaces)





X-ray diffraction analysis



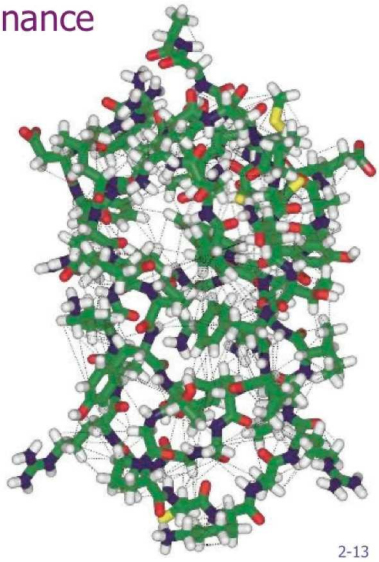
Most exact, but proteins have to be crystallized

Nuclear magnetic resonance

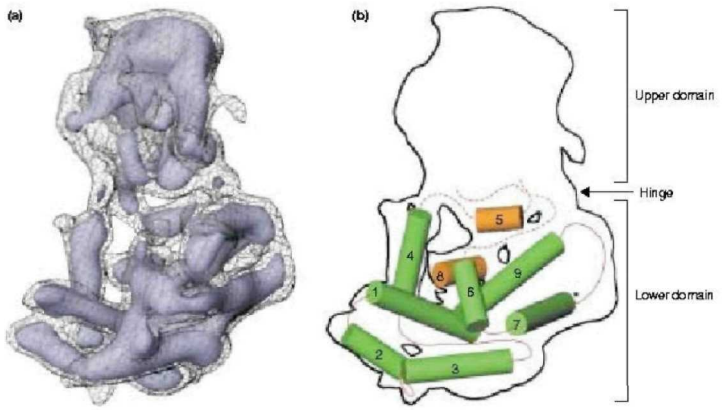
Gives information about hydrogen bonds (spin of electrons)

Proteins in solution

Poor resolution



Determining structure model

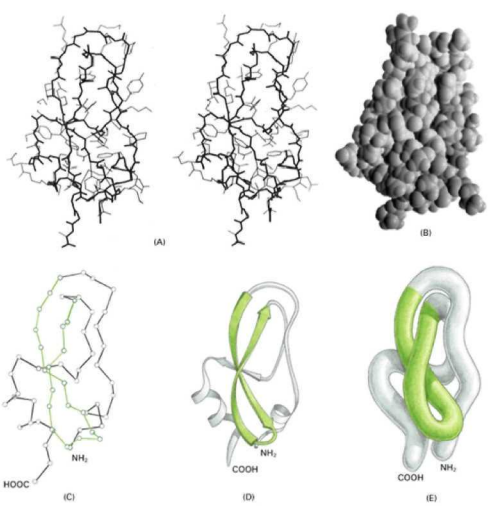


Example: EM density slice of ribosome

2.2

Protein structure visualization

Different visualizations



Representations of 3-D structure

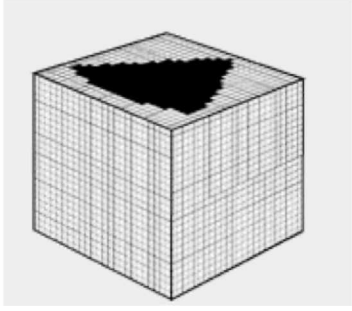
(1) Volume datasets
- Capture electron density directly from imaging methods
- Experimental data
- Direct visualization (volume rendering) difficult

(2) Surface representations
- Isosurfaces of equal density
- Allow better visualization (detail removed, natural appearance with lightging)

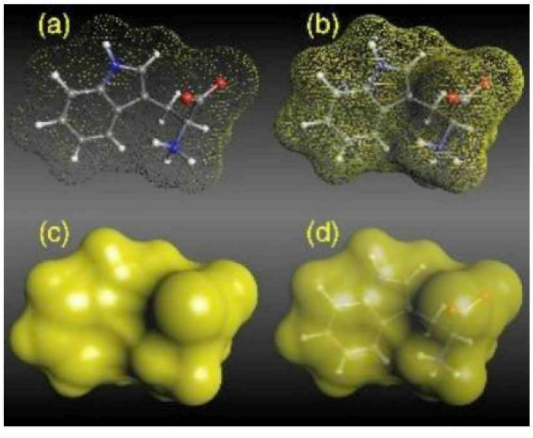
(3) Structure files
- Give direct information about atom positions and secondary structures
- Can be visualized as surfaces or more abstract graphics

Volume dataset

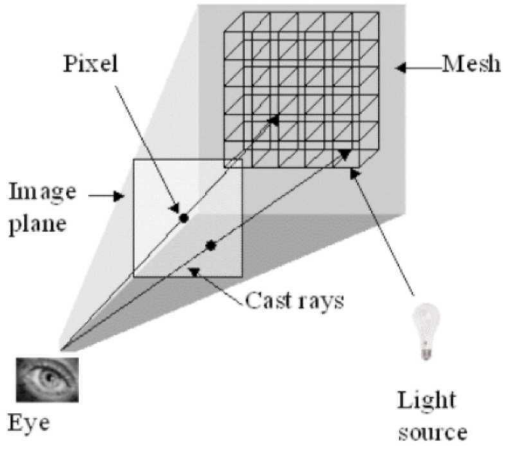
(1) Voxel: (cubic) volume element
(2) Density: scalar value assigned to voxel (grey value)
(3) Three-dimensional array



Surfaces and Structure Models



Rendering: Ray casting



Ray casting: how to determine pixel

(1) Colour and opacity as integral

- over all voxels along the ray

$$\int e^{-\int_0^x \sigma(t)dt} \cdot I(x)dx$$

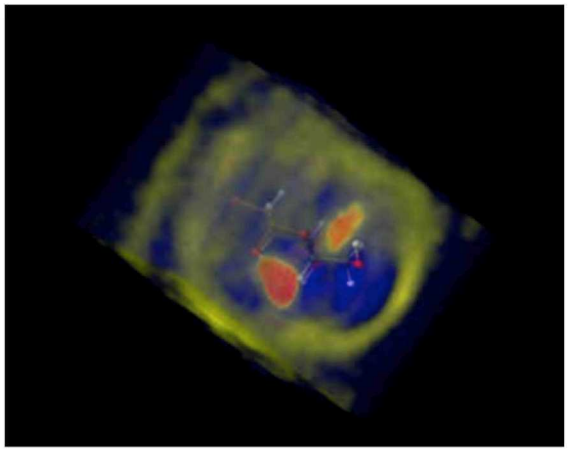
(2) Problem: opacity

- Without opacity you can see only the outside
- Opacity proportional to density: high density regions stick out
- Opacity above a certain threshold: clear surfaces

(3) Simpler mechanisms

- Maximum intensity projection: maximum along ray

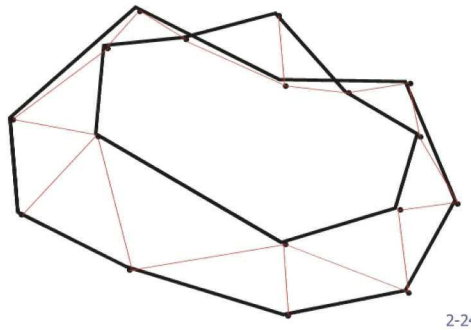
Volume rendering



Triangulation

(1) Surfaces are composed of triangles in 3D
- Rendering uses projection and hidden surface removal plus lighting (angle of triangle to light source)

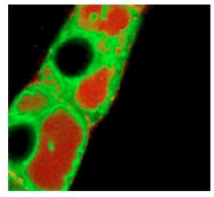
(2) Can be computed from boundaries in a plane connected with triangles



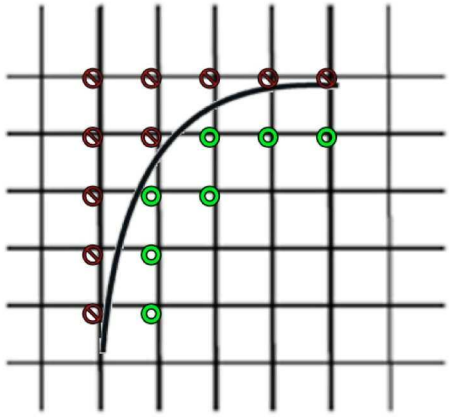
Segmentation

(1) Assignment of each voxel to one of several classes (e.g. object and background)
- E.g., based on a simple threshold
- Or by manual delineation
- Or based on threshold plus connectivity (seedpoints)
- Or based on complex attributes calculated from the environment of a voxel (texture measures)

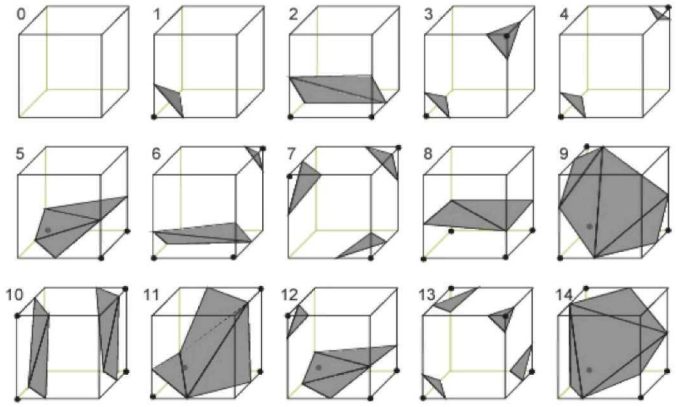
(2) Needed to identify objects within volume datasets



Marching cubes: Surface detection

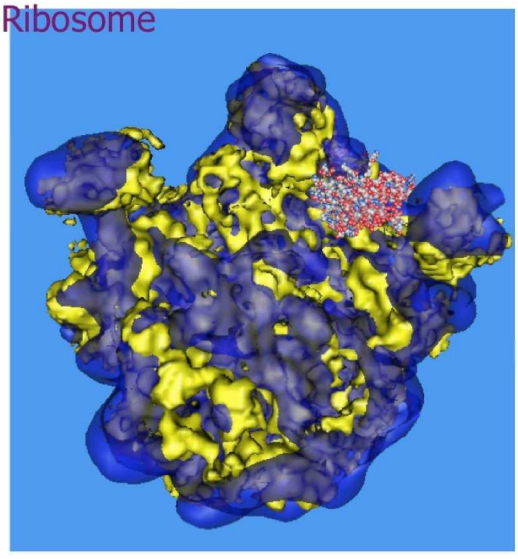


Marching cubes: Surface generation



Surfaces are generated between voxels from different segments

Example: Ribosome



PDB (structure) files

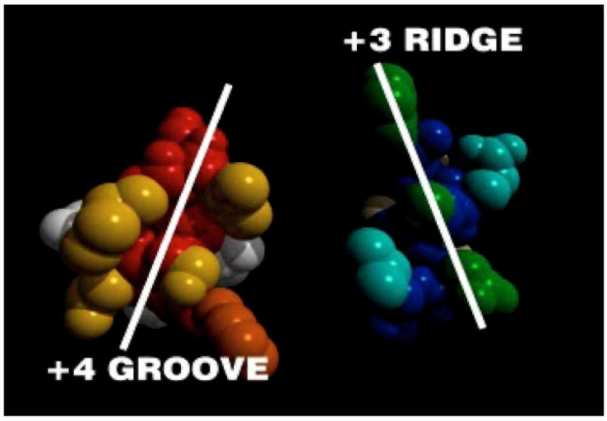
(1) A 3-D coordinate (x,y,z) and binding length for each atom
- Type of binding has to be reconstructed by software („chemistry rules")

(2) „Residue dictionaries" simplify software
- In MMDB (molecular modeling database) type files
- Contains chemical structure of each residue

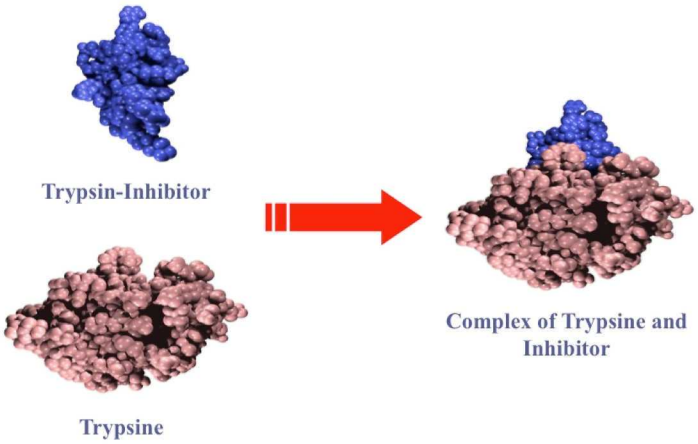
(3) Are available in protein databases
- Swissprot, MMDB, etc.

(4) Visualized as surface graphics by rendering applications
- Such as Chime, RasMol, Swiss PdbViewer

Docking



Docking simulation

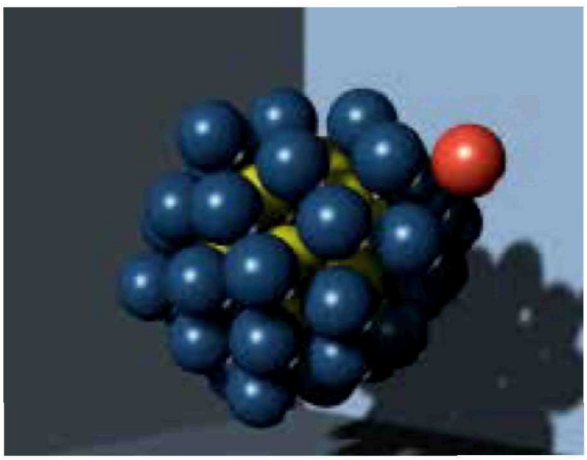


Docking prediction

(1) Generation of candidates
- Rigid structure assumed (simplification)

(2) Calculating fitness function between protein and candidate
- Geometric alignment (collision detection)
- E.g., counting number of van der Waals contacts

Molecular Dynamics



2.3

Protein databases

Protein databases

(1) Sequence and structure
- E.g., SwissProt
- Sequence information as in other sequence databases
- Structure information as PDB (protein data base) file
- Patterns and families

(2) Protein interactions
- E.g., BIND (biomolecular interactions data base)
- Generalized functional information

(3) Accession keys
- Unique identification of protein/database entry
- Should neither change nor carry information
- For reference between entries and databases

SwissProt entry

NiceProt View of SWISS-PROT: P32905

Printer-friendly view | Quick BlastP search

[General] [Name and origin] [References] [Comments] [Cross-references] [Keywords] [Features] [Sequence] [Tools]

General information about the entry	
Entry name	RS0A_YEAST
Primary accession number	P32905
Secondary accession numbers	None
Entered in SWISS-PROT in	Release 27, October 1993
Sequence was last modified in	Release 30, October 1994
Annotations were last modified in	Release 37, December 1998

SwissProt entry (2)

Name and origin of the protein	
Protein name	40S ribosomal protein S0-A
Synonym	Nucleic acid-binding protein NAB1A
Gene name	RPS0A or NAB1A or NAB1 or YST1 or YGR214W
From	Saccharomyces cerevisiae (Baker's yeast) [TaxID: 4932]
Taxonomy	Eukaryota; Fungi; Ascomycota; Saccharomycotina; Saccharomycetes; Saccharomycetales; Saccharomycetaceae; Saccharomyces.

References

[1] SEQUENCE FROM NUCLEIC ACID.
Miles J., Formosa T.G.;
Submitted (MAR-1992) to the EMBL/GenBank/DDBJ databases.

[2] SEQUENCE FROM NUCLEIC ACID.
STRAIN=S288c;
MEDLINE=97435481; PubMed=9290212; [NCBI, ExPASy, EBI, Israel, Japan]
Rieger M., Brueckner M., Schaefer M., Mueller-Auer S.;
"Sequence analysis of 203 kilobases from Saccharomyces cerevisiae chromosome VII.";
Yeast 13:1077-1090(1997).

2-37

Comments

- *FUNCTION*: BINDS DNA. REQUIRED FOR THE ASSEMBLY AND/OR STABILITY OF THE 40S RIBOSOMAL SUBUNIT.
- *MISCELLANEOUS*: THERE ARE TWO GENES FOR S0 IN YEAST.
- *SIMILARITY*: BELONGS TO THE S2P FAMILY OF RIBOSOMAL PROTEINS.

Copyright

Cross-references	
EMBL	M88277; AAB05643.1; -. [EMBL / GenBank / DDBJ] [CoDingSequence] Z72999; CAA97241.1; -. [EMBL / GenBank / DDBJ] [CoDingSequence]
PIR	S42143; S42143.
SWISS-2DPAGE	P32905; YEAST.
SGD	S0003446; RPS0A.
GeneCensus	P32905; YGR214W.
InterPro	IPR001865; Ribosomal_S2. Graphical view of domain structure.
Pfam	PF00318; Ribosomal_S2; 1.
PRINTS	PR00395; RIBOSOMALS2.
TIGRFAMs	TIGR01012; Sa_S2_E_A; 1.
PROSITE	PS00962; RIBOSOMAL_S2_1; 1. PS00963; RIBOSOMAL_S2_2; 1.
ProDom	[Domain structure / List of seq. sharing at least 1 domain].
BLOCKS	P32905.
ProtoMap	P32905.
PRESAGE	P32905.
DIP	P32905.
ModBase	P32905.

2-38

SwissProt entry (4)

Keywords

Ribosomal protein; DNA-binding; Acetylation; Multigene family.

Features

Key	From	To	Length	Description
INIT_MET	0	0		
MOD_RES	1	1		ACETYLATION.

 Feature table viewer

Sequence information

Length: 251 AA Molecular weight: 27893 Da CRC64: 4FF263575B82C75A [This is a checksum on the sequence]

```
        10         20         30         40         50         60
         |          |          |          |          |          |
SLPATFDLTP EDAQLLLAAN THLGARNVQV HQEPYVFNAR PDGVHVINVG KTWEKLVLAA

        70         80         90        100        110        120
         |          |          |          |          |          |
RIIAAIPNPE DVVAISSRTF GQRAVLKFAA HTGATPIAGR FTPGSFTNYI TRSFKEPRLV

       130        140        150        160        170        180
         |          |          |          |          |          |
IVTDPRSDAQ AIKEASYVNI PVIALTDLDS PSEFVDVAIP CNNRGKHSIG LIWYLLAREV

       190        200        210        220        230        240
         |          |          |          |          |          |
LRLRGALVDR TQPWSIMPDL YFYRDPEEVE QQVAEEATTE EAGEEEAKEE VTEEQAEATE

       250
         |
WAEENADNVE W
```

P32905 in FASTA format

2-39

BIND entry

Interaction

Interaction ID: 6155

Accession date: Sep 5, 2001
Description: Tyrosine phosphorylated Gab1 recruits PI3K by direct interaction with the p85 subunit

Molecule A

Gab1
Description: Grb2-Associated Binder-1. A docking protein that contains a PH domain, several proline-rich stretches and multiple tyrosine phosphorylation sites which are SH2
Molecule Type: Protein
GI: 4503851 (NCBI) (SEQHOUND) (BIND)
Molecule origin: Organismal
Organism: Homo sapiens

BIND entry (2)

Molecule B

PI3K p85-alpha
Description: Phosphatidylinositol 3-kinase, p85 subunit {alpha}.
Molecule Type: Protein
GI: 105122 (NCBI) (SEQHOUND) (BIND)
Molecule origin: Organismal
Organism: Homo sapiens

Visualize Interaction!

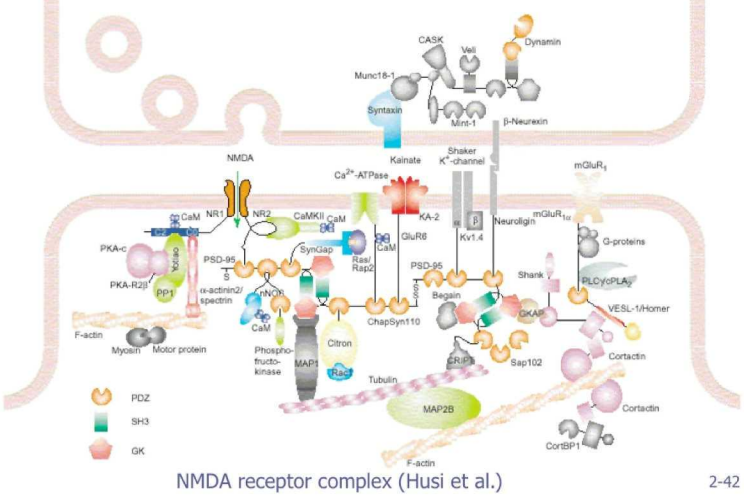
View other information?

Main Info	Publications	ASN.1 XML
Cellular Place	Experimental Condition	Conserved Sequence
N/A	N/A	N/A
Binding Sites	Chemical action	Chemical State
N/A	N/A	N/A

Comments and suggestions to: < info@bind.ca >

BIND Homepage

Protein complexes



NMDA receptor complex (Husi et al.)

2.4

Protein structure prediction

Protein structure prediction

(1) Homology-based
- Fragment approaches: finding aligned segments without gaps (SCR structurally conserved regions, e.g., to identify helices or sheets from known sequences) and loops (SVR structurally variable regions) to match other known segments with least variation

(2) Threading
- Generation and checking of many different (rough) alignments
- Scoring, e.g. with probability distribution of distances between amino acid pairs (position and sequence number)

(3) Conformational energy
- Generate conformations
- Minimize energy function
- Not efficient!/Local minima!

Metropolis algorithm

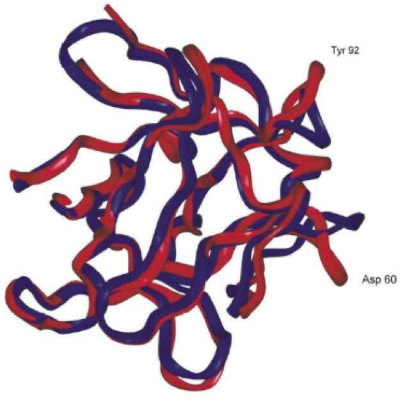
(1) Start
- Any molecule conformation x with energy E (x)

(2) Loop
- Randomly disturb x into x‘
- If E (x‘) < E (x) continue loop with x‘
- Else randomly choose whether to continue with x or x‘ (probability depending on a virtual temperature T)

(3) Can escape local minima
- If temperature is high, randomly move away from local minima with some finite probability

(4) Simulated annealing
- Start with a high temperature
- Slowly lower the temperature to move into minimum
- Guaranteed success with slow temperature reduction
- But usually faster reduction required

Example structure prediction



Red is prediction by SwissModel (RMS = 1,54Å)

2.5

Protein identification by mass spectrometry

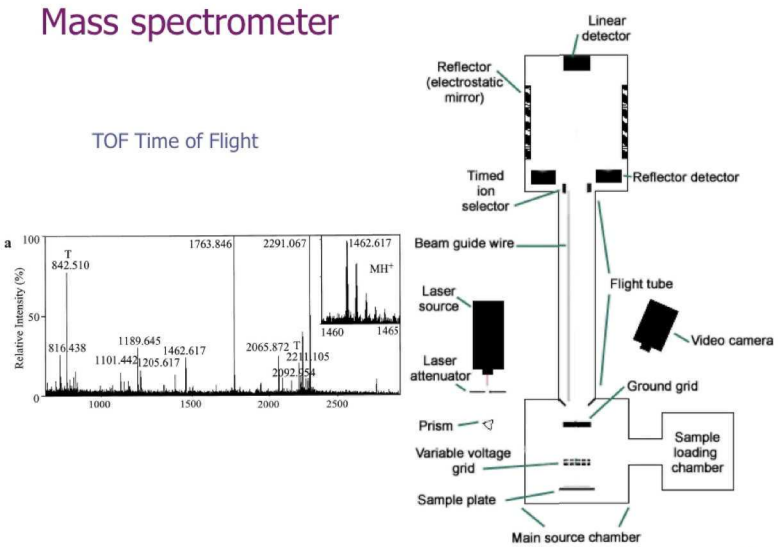
Mass spectrometry

(1) Mass spectrometry separates according to mass

- E.g., TOF (time of flight): peptides are ionized and accelerated in an electric field
- Detector produces a peak for several flight times, flight time correlates with mass

Mass spectrometer

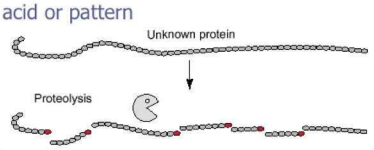
TOF Time of Flight

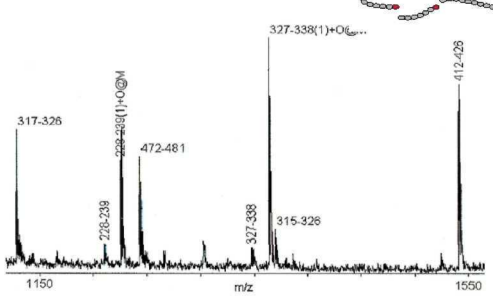


Peptide mass fingerprinting

(1) Proteins are split into peptides by sequence-specific proteases

- Always break at the same amino acid or pattern
- For a given sequence, a unique set of peptides results
- Mass of all peptides is measured





Peptide mass fingerprinting

(1) Database search
- Mass spectrum is compared with theoretical spectra for proteins in a database
- Best match is obtained

(2) Match
- Peak from spectrum equals one theoretical peptide
- Within mass tolerance/accuracy
- Multiple matches from the spectrum possible (counts only as one)

(3) Shared peak count
- Highest number of matches

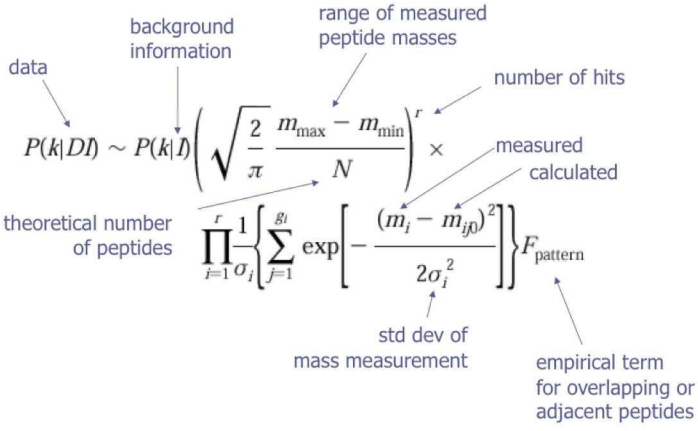
Problems with shared peak count

(1) Not all peptides occur in the mass spectrum
- low abundance, further fragmentation, incomplete ionisation, phosphorylation, etc.

(2) Is the protein really in the database?
- spurious matches with proteins from other organisms, mutations

(3) Long proteins tend to be preferred
- as the are more opportunities for matches

Statistical approach: ProFound



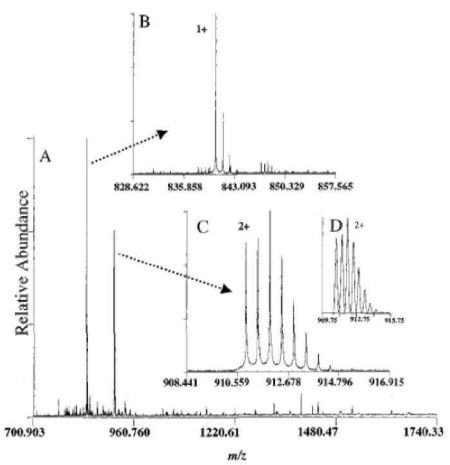
$$P(k|DI) \sim P(k|I)\left(\sqrt{\frac{2}{\pi}}\,\frac{m_{\max}-m_{\min}}{N}\right)^{r} \times \prod_{i=1}^{r}\frac{1}{\sigma_i}\left\{\sum_{j=1}^{g_i}\exp\left[-\frac{(m_i-m_{ij0})^2}{2\sigma_i^{\,2}}\right]\right\}F_{\text{pattern}}$$

Single peptide identification

(1) Experimental constraints should make database search result unique

- High resolution (high accuracy) MS restricts hits
- Cysteine-containing peptides only (rare):
  Cysteine tag removes all other peptides
- Known organism (limited number of genes/proteins)

High resolution mass spectrometry



Example: Isobaric peptides in yeast

protein name[a]	protein mass[b]	peptide isobars	peptide mass	error (ppm)
laminin fragment	2 426.271	***R***YVVLP***R***PVCFE***K***	1604.886[c]	0.189
TPI1	26 778.962	FLAS***K***LGD***K***AASEL***R***	1604.889	1.747
YPR143W	52 661.268	D***KKR***L***RK***NAEFG***R***	1604.886	0.07
YDR428C	52 785.179	NLYDAVSNIT***R***LV***K***	1604.889	1.747
TAP42	31 135.369	IELFQ***R***N***K***EIST***K***	1604.889	1.747
CYC2	37 692.826	VQL***K***IFETD***R***QT***K***	1604.889	1.747
MFS1	46 151.311	ESHPVGIL***R***DLIE***K***	1604.889	1.747
YMR291W	64 850.898	EFDLL***R***SISE***K***L***R***	1604.889	1.747
YKR078W	51 708.695	L***R***TAEDEY***R***VL***K***	1604.889	1.747
TFC6	75 311.526	D***K***IE***K***YGLN***K***E***K***	1604.889	1.747
SSE1	77 318.483	YLA***K***EEE***KK***QAL***R***	1604.889	1.747
YBR102C	85 484.685	LDEFI***KK***NSD***K***L***R***	1604.889	1.747
STB6	88 779.841	***K***ISADLN***K***IDGLY***R***	1604.889	1.747
FZ01	97 746.957	E***K***NGFNIE***KK***ALS***K***	1604.889	1.747
SEC10	100 279.455	NES***K***IV***KR***VFEE***K***	1604.889	1.747
YLL005C	102 103.872	L***K***ELLFELFYY***K***	1604.885	0.234
S51441	105 161.643	HTVTEL***K***SEIHAL***K***	1604.889	1.747
PEX1	117 202.758	EEV***K***DIIE***K***HLP***K***	1604.889	1.747
RRP5	193 015.955	A***K***D***KKK***VEDLFE***R***	1604.889	1.747
DOP1	194 565.002	LTSSLSPALPAGVHQ***K***	1604.889	1.747

Only one containing cysteine

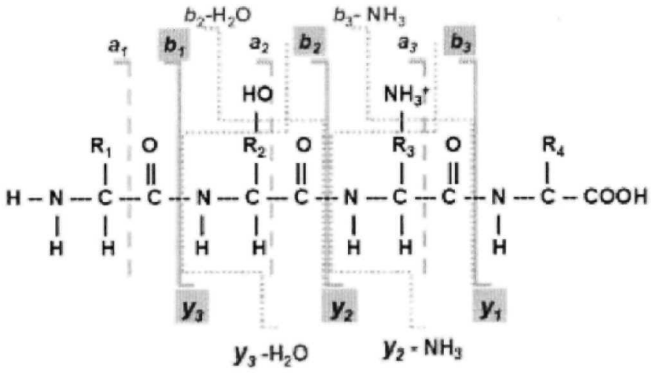
Distinction also possible through protein mass

MS/MS

(1) Subject peaks to a further MS step
- Breaking each peptide (mechanically) into ion fragments
- Measuring fragment spectrum
- Identify peptide sequence through comparison with theoretical fragmentation

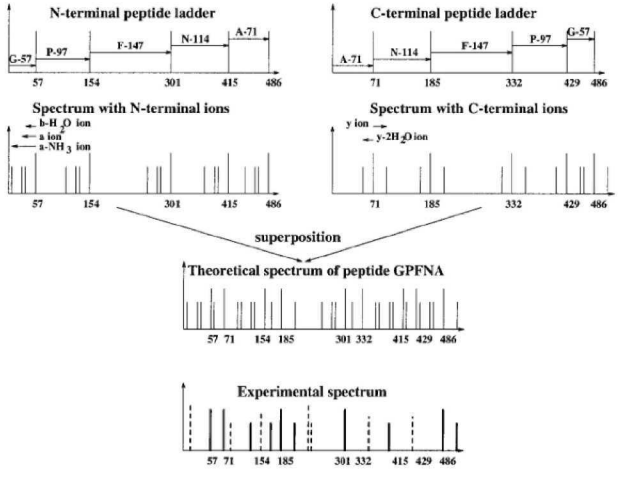
(2) For unknown organisms
- Can search among data from all organisms and even partial sequences

Typical ions



Mostly C-terminal or N-terminal ions (few middle fragments)

Experimental vs theoretical spectrum



Peptide identification problem

(1) Peptide P
- Sequence of amino acids $p_1 \ldots p_n$
- Mass m (p) = sum (m (p_i))

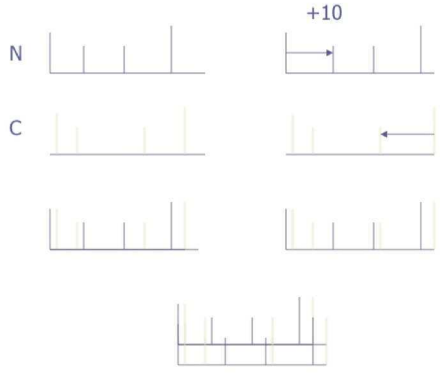
(2) Ion types
- $\{d_1,..,d_k\}$ numbers (weight difference)
- d-ion of partial peptide P has mass m (P) - d

(3) Spectrum
- $S = \{s_1,..,s_n\}$ is a set of (experimental) masses

(4) Problem:
- Given S, ion types and mass m
- Find a peptide of mass m with maximal match to S
- Scoring function can be shared peak count or more probabilistic

The effect of mutations

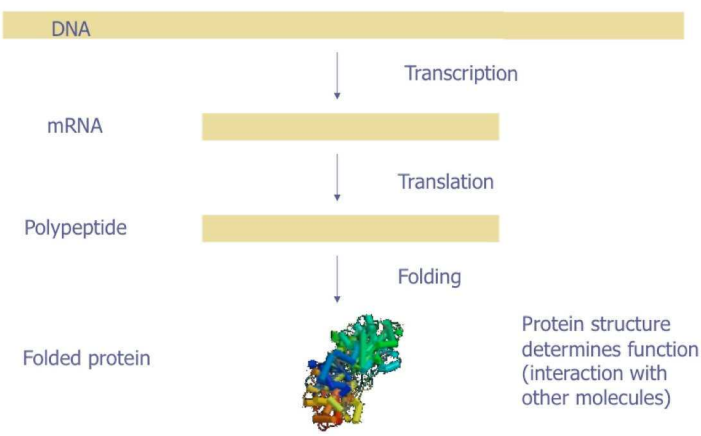
(1) A single mutation halves the shared peak count



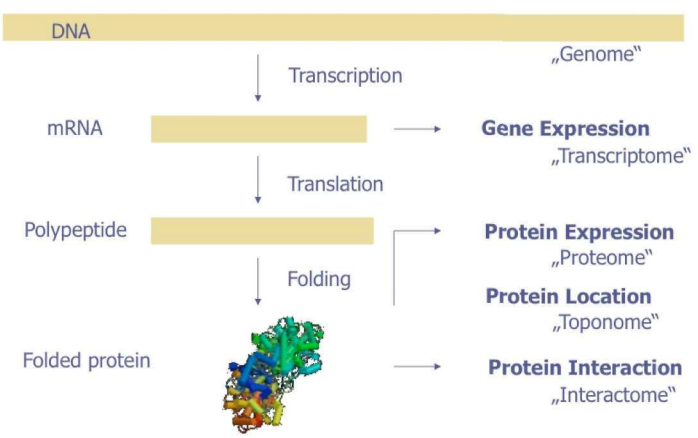
Part 3

Protein Expression and Function

Gene to proteins



Expression analysis



Gene expression and cell specialization

(1) Expression is different in different cell types
- Same genome
- Each cell type produces the proteins it needs
- Expression differences reveal function of the different gene products

(2) Expression differences may indicate pathology
- E.g., cancer cells show expression changes to normal cells
- Is of diagnostic value (might discover cancer at the cell level)
- Shows functional mechanisms of cancer (which proteins are responsible for the new behaviour)

Expression and dynamics

(1) Expression is a dynamic process
- Expression levels vary over time
- Internal regulation, e.g., in cell cycle control
- External regulation in response to signals or changing conditions

(2) Expression regulation is robust
- Cell function can tolerate fluctuations in expression levels
- Thus even large differences in expression do not necessarily imply different function

(3) Expression takes time
- Protein expression may take minutes or hours
- Isolated snapshots might be misleading

Expression snapshots

(1) Expression measurement must be „quick"
- To get an accurate expression profile at time t, cell processes have to be interrupted
- Otherwise, chemical reactions will continue in an uncontrolled manner before the actual reading can be taken

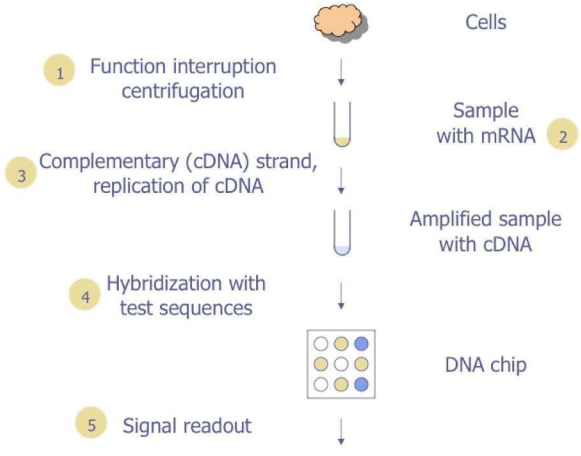
(2) Sample preparation is crucial
- Usually, many cells are taken together and are destroyed, e.g., by centrifugation
- Processes at a time level of seconds and even minutes might still happen (e.g., stress response to centrifugation)

(3) Multiple snapshots are needed
- Over time
- Over different external conditions

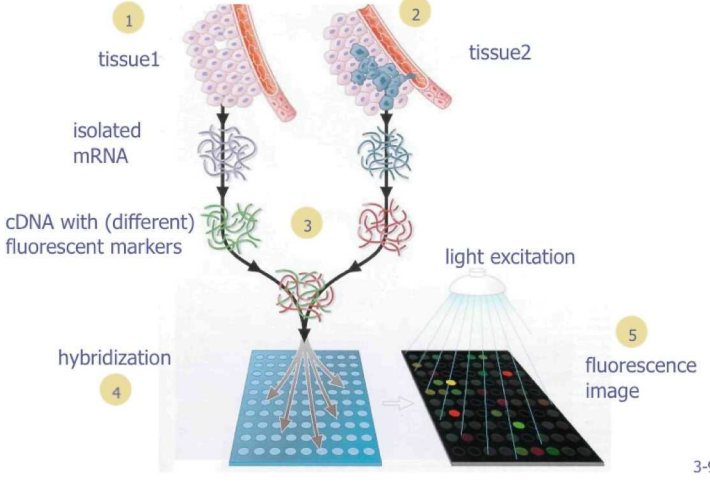
3.1

Gene expression analysis

Gene expression experiments

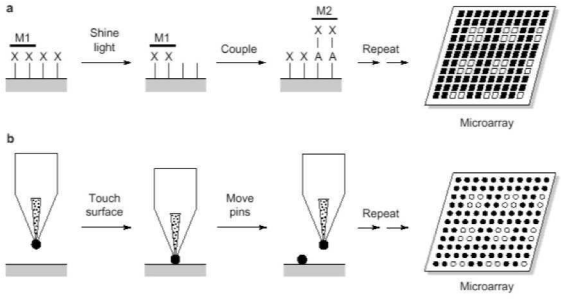


Hybridization experiment



Microarray technologies

Photolithography (X = photocoating, stepwise by base)



Liquid spotting (of complete DNA molecules)

Photolithography

(1) On-chip synthesis of DNA

- An optical mask selectively removes light sensitive coating
- Free places are flooded with one base
- Enzyme couples it to previous bases
- Need 4 masks (4 bases) per element: expensive and time-consuming!
- Can be directly generated from a database

(2) Affymetrix chips

- Short oligonucleotides (25 base pairs) per spot
- Because of long production time
- Longer sequences are matched through multiple oligonucleotides
- Software needed for correct oligo design, so that each gene has about 16 or more specific matching oligos

(3) Alternative: Ink-jet printing

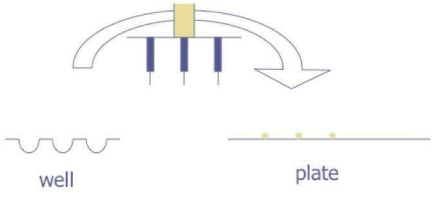
- Piezoelectric (focused) spraying of individual bases

Liquid spotting

(1) Robotic pipetting from a library

- On glass substrate or nylon filters
- Each sequence needs to be synthesized
- Problem of contamination
- Problem of correct synthesis

(2) Equipment is already present in most labs



Hybridization

(1) Need to get rid of ribosomal RNA
- Can be 80% of total cell RNA

(2) cDNA is tagged with fluorescent dye
- Two samples can be mixed with different colours
- Radioactive labeling also possible

(3) Sample cDNA attaches to probe sequence
- Hybridization errors: may also attach to similar sequences

(4) Two probes for each sequence
- Perfect match (PM): identical sequence
- Mismatch (MM): one different base in the middle
- Actual signal is difference between perfect match and mismatch (sort of background noise from a number of similar sequences)

Readout

(1) Laser excitation
- Focus sequentially on individual pixels to avoid background noise from scattered light
- Optical detector readout
- Many pixels per spot = slow

(2) Light excitation
- Simultaneous lighting of whole chip
- High resolution camera
- Fast, but lower signal-to-noise ratio
- Possible for lower density arrays

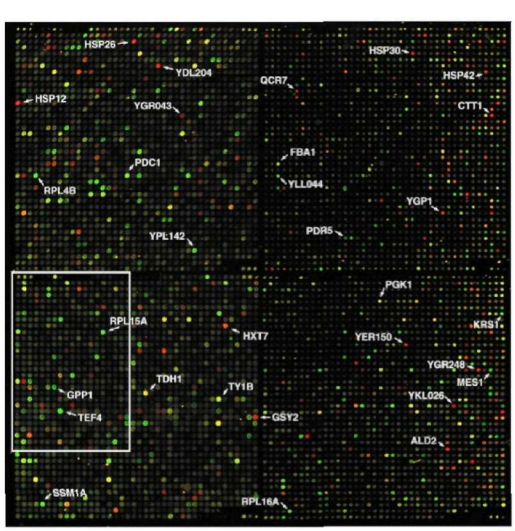
(3) Phosphor imager
- For radioactive readout

Example

6400 yeast DNA sequences printed on a glass slide 18 x 18 mm

Two samples were prepared with read and green dyes, respectively

From: deRisi et al, Science 278, 1997, 680-6.

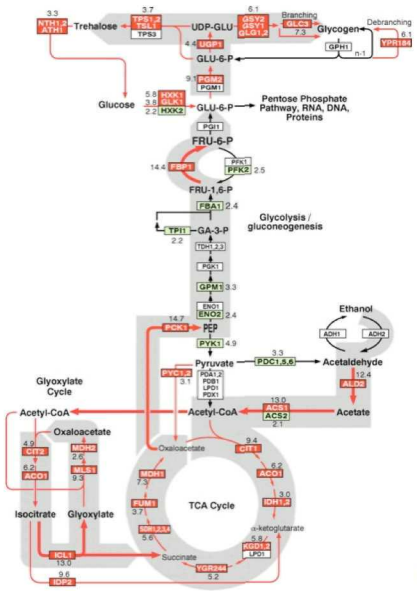


Diauxic shift

Yeast shifts from producing ethanol to consuming ethanol when glucose is depleted

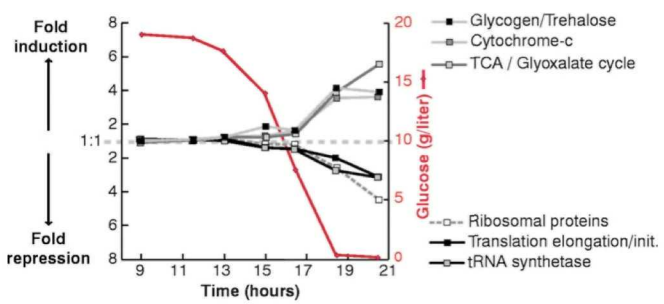
Red: Increased expression

Green: Decreased expression



From: deRisi et al, Science 278, 1997, 680-6.

Results from the expression experiment



From: deRisi et al, Science 278, 1997, 680-6.

3.2

Proteomics

Proteins after Translation

(1) Degradation
- To create equilibrium between synthesis and degradation

(2) Posttranslational modifications
- Individual amino acids are modified (e.g., phosphorylation)
- These modifications may activate/deactivate function

(3) Transport
- Into different parts of the cell or outside the cell

(4) Assembly of protein complexes
- Often required for protein function

Proteomics

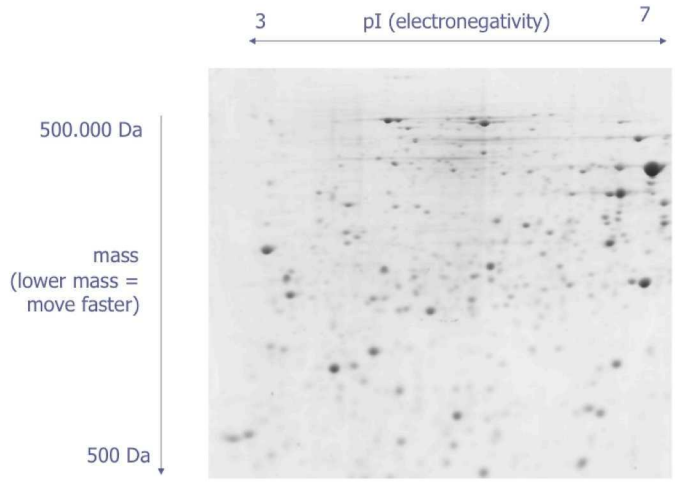
(1) Quantitative analysis (expression proteomics)
- Gel electrophoresis (old)
- Quantitative mass spectrometry of protein mixtures (new)

(2) Posttranslational modifications
- Mass spectrometry to detect weight differences

(3) Localization (topological proteomics)
- Selective purification of organelles
- Microscopy with fluorescent markers

(4) Complex assembly (interaction proteomics)
- Affinity purification of complexes through baits
- Bilateral interaction (yeast two hybrid screen, protein chips)

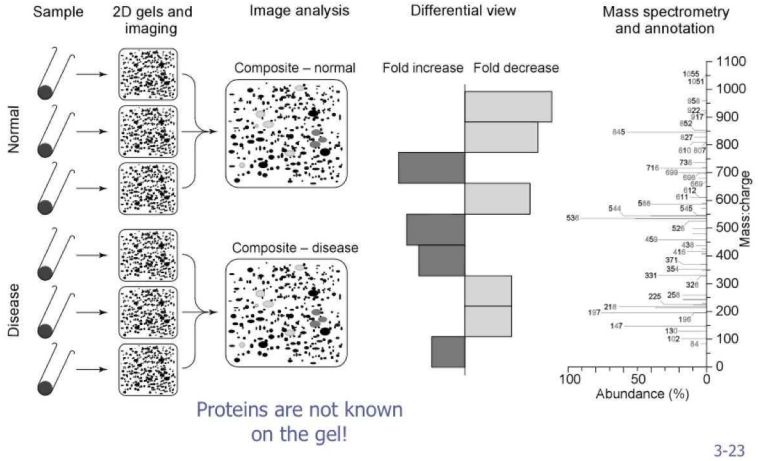
Gel electrophoresis



Electrophophoresis preparation

(1) Sample preparation
(2) Electrostatic separation on strip (pI)
(3) Mass migration in gel chamber (vertical)
(4) Staining (silver or fluorchromes)
(5) Scanning (flatbed or camera)

Electrophoresis experimental approach



Problems in Gel Electrophoresis

(1) Reproducibility

- Spot intensities and locations vary even for identical gels created in the same processing runs.
- Not all proteins easily make it onto the gel, and different proteins stain in different ways, possibly in a non-linear relation to concentration. Thus, concentrations of different proteins are not easily comparable.
- Not all proteins are chemically stabilized so that ongoing reactions cause a smearing of certain proteins over a whole area.

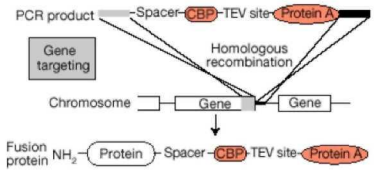
(2) BUT

(3) Electrophoresis is the only technique that allows quantification of a large number of proteins at the same time

Affinity purification

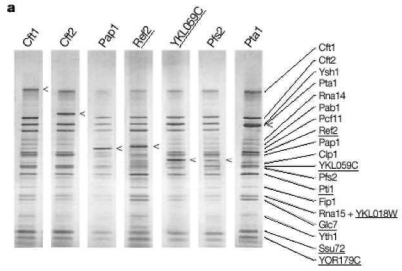
(1) Reduce the number of proteins and image purity
- Through selecting only a subset of proteins from the sample
- E.g., selective centrifugation
- E.g., membrane only

(2) Bait
- (Genetically) tagging particular proteins (e.g. in yeast)
- Selecting from the sample (gently) all proteins that couple to this protein (partial complexes)

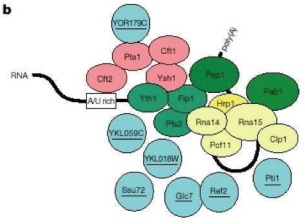


From: Gavin et al., Nature 415, 2002, 141-7

Polyadenylation machinery in yeast



Similar 1D bands from several related baits

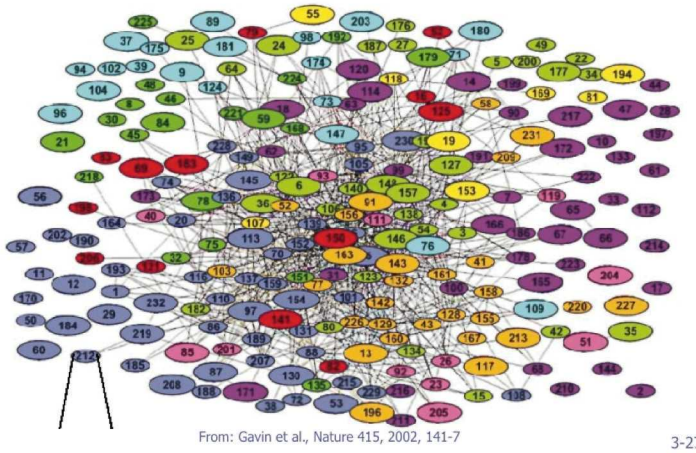


Proposed polyadenylation machinery

From: Gavin et al., Nature 415, 2002, 141-7

Protein complexes in yeast

Links indicate shared proteins



From: Gavin et al., Nature 415, 2002, 141-7

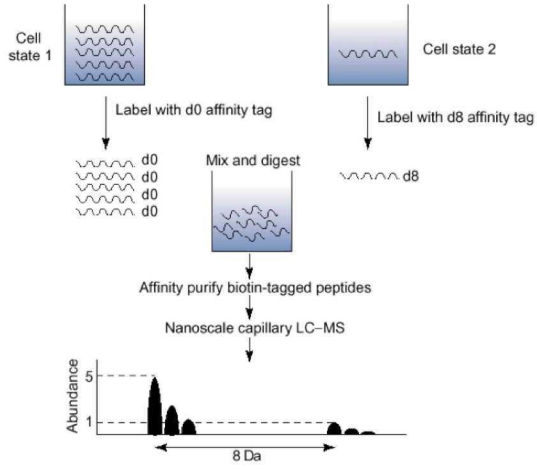
Protein mixtures by MS

(1) Each protein in the mixture is identified by a single tagged peptide
- MS/MS identifies peptide sequence and thus the identity of the parent protein
- Requires constrained protein mixture

(2) Mixture is further separated by liquid chromatography (LC)
- Mass-based separation of peptides with direct input into MS

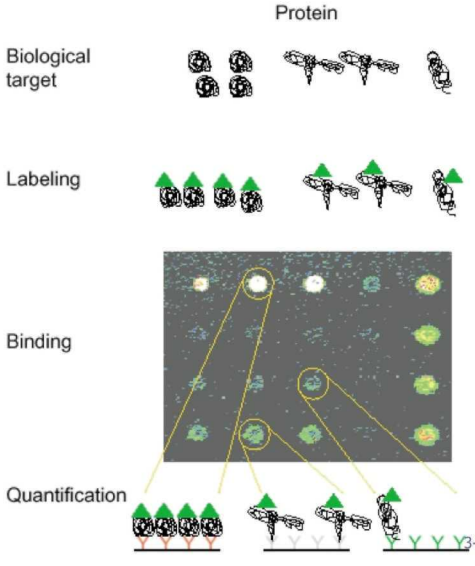
(3) Relative abundance by peak difference between two different tags
- Different weight tags for test and reference sample

Isotope-coded affinity tagging (ICAT)

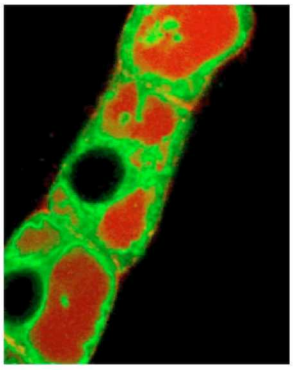


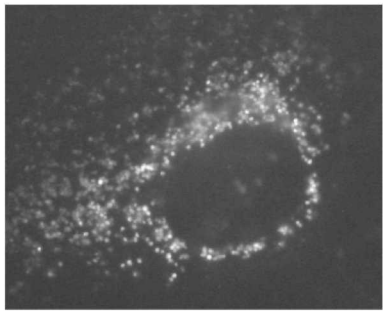
Protein chips

Require specific knowledge about selectivity and specificity of antibodies on the chip



Protein localization by microscopy





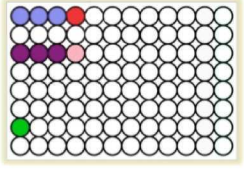
Proteins labeled by fluorochromes

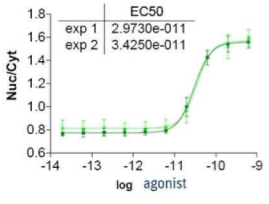
Toponomics

- Topological Proteomics
  - *Measuring* spatial distribution of some/all proteins
  - Also called topoproteomics/location proteomics
- Toponomics
  - Toponome gives the laws of spatial arrangement[1] (not necessarily causal)
  - *Modeling* spatial distribution based on measurements
  - Reducing observed spatial distribution into representative descriptions

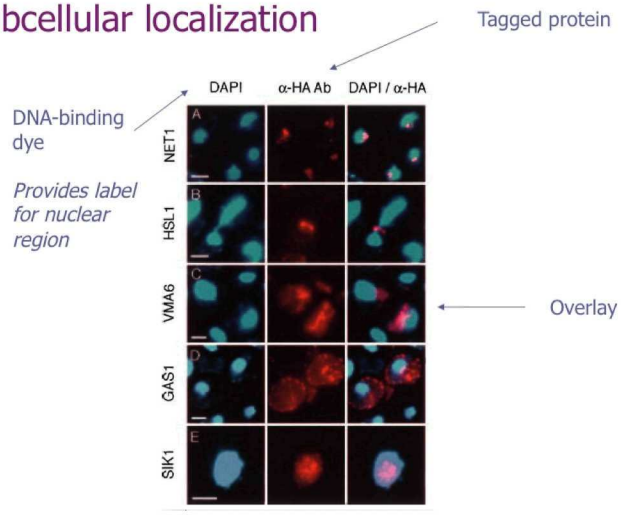
High Content Screening

- Automated microscopy of cellular events
  - Input: System perturbation
  - Output: Population response
- Screening = Optimization of input
  - *Using* toponomics model to derive numerical descriptor
  - E.g., dose-response curve
  - Black-box model with respect to molecular mechanism



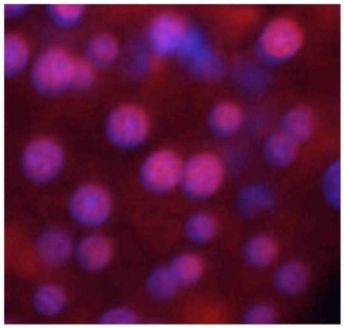


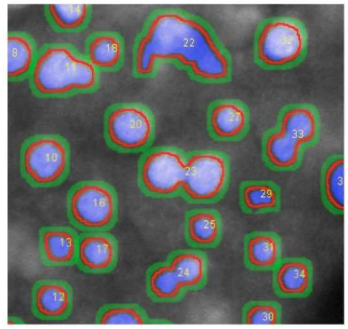
Subcellular localization



From: Kumar et al., GENES & DEVELOPMENT 16(2002):707 –719

Nuclear translocation

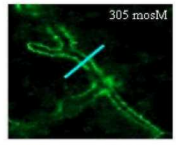


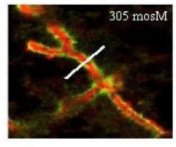


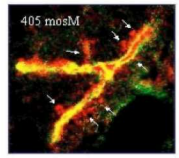
1. Nuclear region segmentation (adaptive threshold)
2. Region extension
3. Quantification

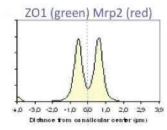
Toponomics of transport

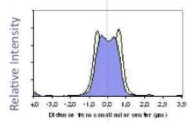
- Dual labels
  - → Topological markers: ZO1 is a tight junction protein, indicates structure
  - → Functional markers: Transport protein is regulated through translocation
  - → Spatial distribution (profiling functional marker orthogonal to structure) gives information about regulation state

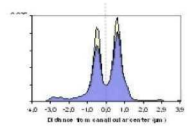






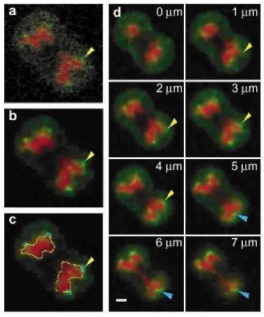




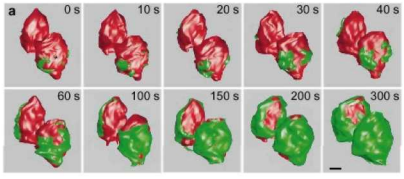


Dynamic 3-D imaging

Image slices with two different fluorochromes (red and green)



3-D reconstruction of isosurfaces in a temporal sequence



From: Gerlich et al., Nature Cell Biology 3(2001):852-855

3.3

Image processing

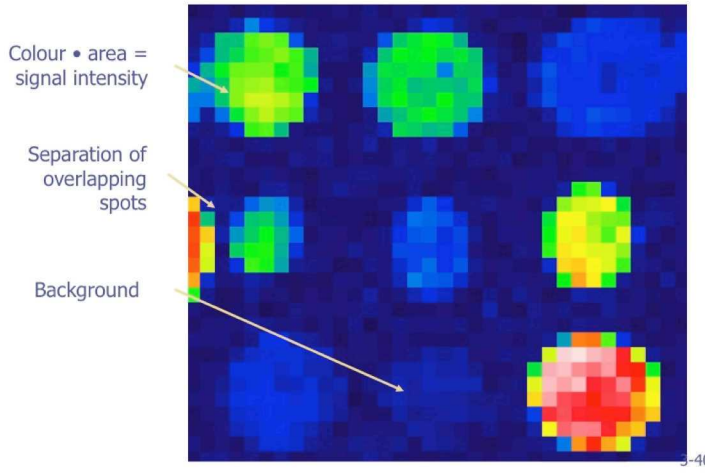
Steps in image processing

(1) Segmentation
- Distinguishing signals from background
- Distinguishing separate signals

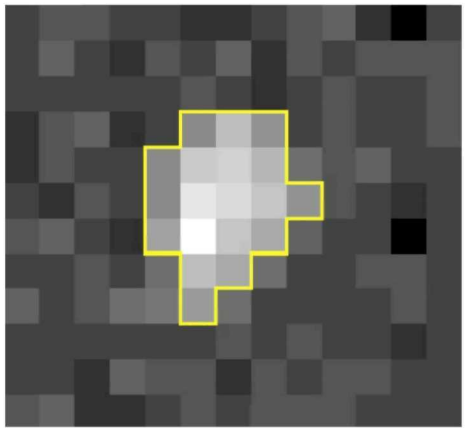
(2) Quantification
- Estimating signal intensity

(3) Background estimation/removal
- Subtracting non-uniform background signals
- Raw materials under ambient lighting of the excitation source/scanner light

Magnified subsection



Irregular segmentation



Segmentation

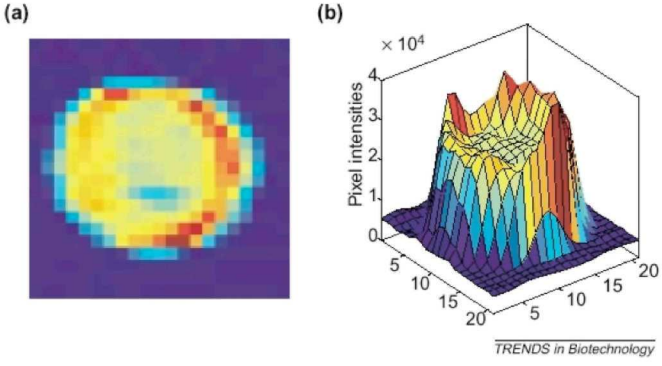
(1) Assigning each pixel to a particular class
- Here typically spot or background
- Each separate spot can be its own class

(2) Simple thresholding algorithm
- Given an intensity threshold t
- For all pixels (i, j) in ascending coordinate order:

intensity (i, j)	class (i, j-1)	class (i-1, j)	NEW class (i, j)
<=t			0
>t	0 or k	0 or k	k
	k	m	join k+m

(3) Problem: choosing a threshold

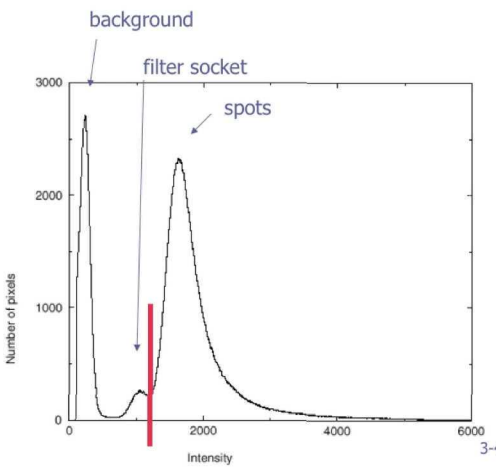
Spot detail



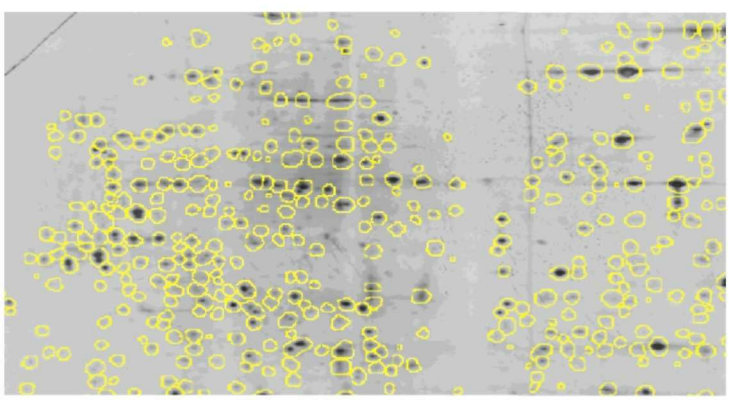
Pixel intensities coded as colours

Histogram thresholding

Finding and counting minima and maxima in the image histogram



Example segmentation



Quantification

(1) Average peak

- Average over all segmented pixels
- Does not take into account size variations, only for uniform spots
- Background-sensitive: non-uniform background will distort values

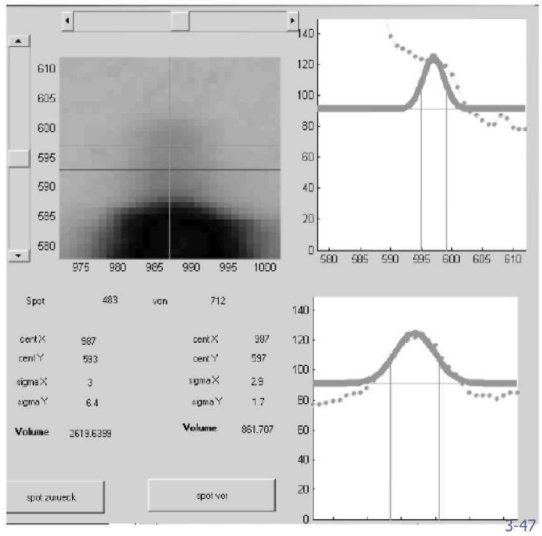
(2) Total intensity

- Summing up intensities of all segmented pixels
- Background-sensitive
- Segmentation-sensitive: unless spots have sharp borders, small signals vary considerably with segmentation threshold

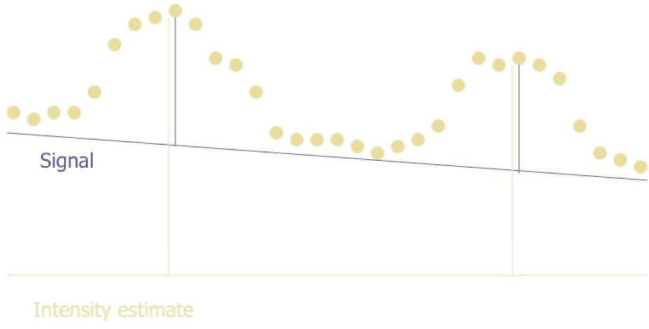
(3) Curve fitting

- Calculating model curve with smallest distance from segmented data
- Noise-sensitive: outliers can skew fitting

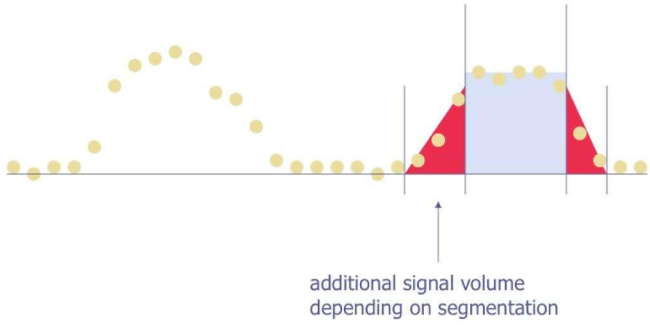
Spot shape



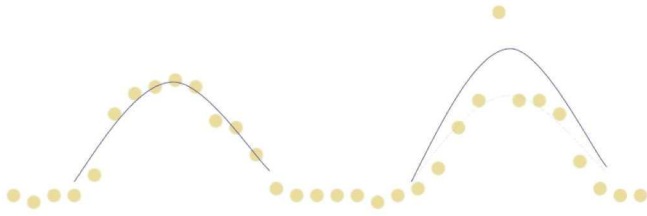
Background sensitivity



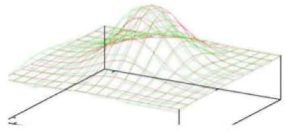
Segmentation sensitivity



Noise sensitivity



Gauss fitting



$M_R = \{p\ (x, y) \mid (x, y) \text{ segmented}\}$

$M_v\ [H, B, c_x, c_y, s_x, s_y] = \{G\ (x, y) \mid (x, y) \text{ segmented}\}$

$$G(x, y) = H \exp[-(x-c_x)^2/2s_x^2] \exp[-(y-c_y)^2/2s_y^2] + B$$

H	Height	**Model Instantiation**
B	Background	
c_x c_y	Center	
s_x s_y	Standard Deviation	

Least-squares Fit of M_R and M_V with approximate start values

Background estimation/subtraction

(1) Background is not uniform
- Different lighting conditions in different part of the chips, optical non-uniformities
- No single threshold for the whole image!
- Not only relevant for threshold, but background (lighting) has to be subtracted as well to get real signal intensity

(2) Background estimation (after segmentation)
- Average of surrounding non-spot areas
- Fitting

(3) Background subtraction (before segmentation)
- Morphological operators

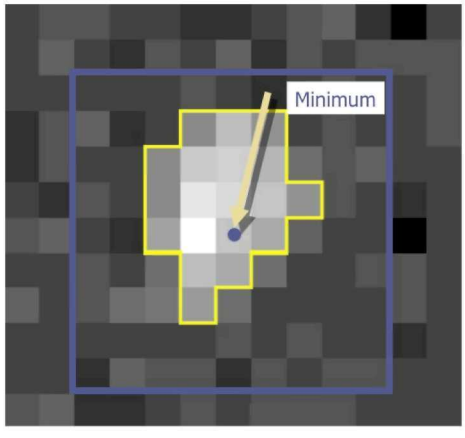
Morphological opening

(1) Pixel substituted by minimum over neighbourhood
- Shape of the neighbourhood is called the *structural element*
- Size of the structural element determines smoothness of operator

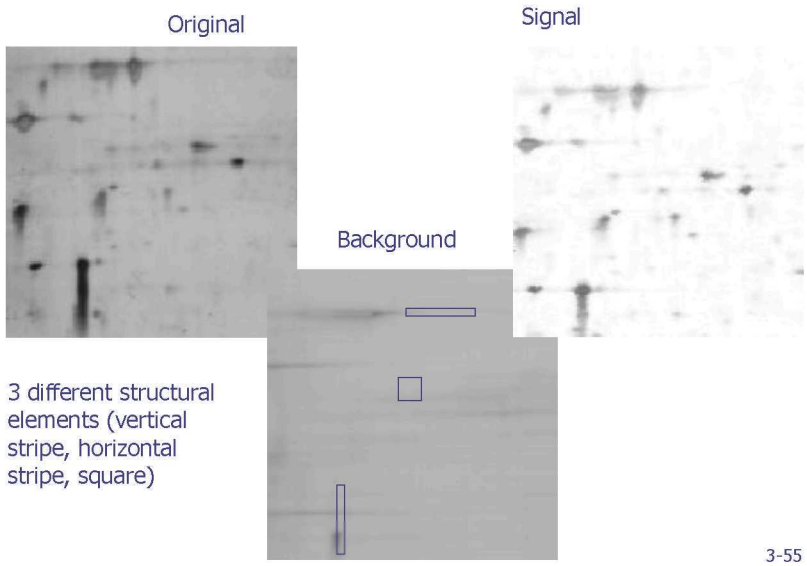
(2) Creating a background image
- Underestimating the background
- Structural element must be larger than spot size, then there will always be background pixel in the region

(3) Creates more reproducible quantitative results than local background estimation

Rectangular element



Example



3-55

Registration

(1) Mapping between different images
- Mapping of different coordinate systems that have a common reference
- E.g., same grid of spots, but distorted carrier/image

(2) Rigid registration
- Translation and rotation only
- Assume same scale
- E.g., different scanning positions

(3) Elastic registration
- Eliminate distortion
- E.g., distortion in different optical systems
- E.g., mechanical elasticity of carriers
- E.g., spatial variations in biophysical processes

3-56

Registration to spot grid

(1) Orientation
- Manually by clicking on border spots
- Automatically (depending on particular features, like recognizable bord angles, markers, etc.)

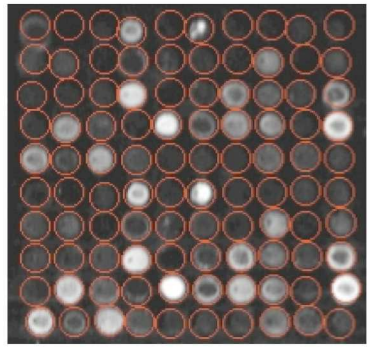
(2) Identification of raster positions
- Because of imprecise needle positions and depositing

(3) Comparatively simple because size of raster is known

Example registration

1. Seed points at theoretical grid points
2. Calculate center of gravity
(Sum of positions x intensity divided by total intensity)
3. Smallest circle above threshold

Mapping function assigns each circle to corresponding grid point



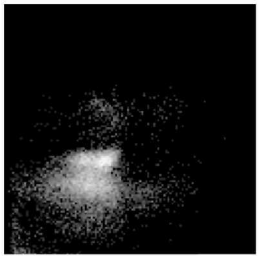
Gray level correlation

(1) Joint histogram
- p (a, b): Number of pixels where intensity a in the first image maps to intensity b in the second image
- Maximize correlation coefficient
- Requires same colour scale

(2) Joint entropy
- - sum (p (a,b) log p (a,b))
- Is higher if there are few pronounced entries in the joint histogram
- Works also with different colour scales (e.g. white matches with dark red, gray matches with light red, black matches with green)

(3) Optimize mapping (rigid, non-rigid) for maximum correlation

Example joint histogram

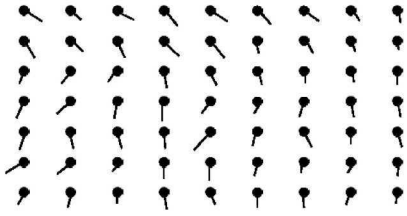


Unmatched

Matched

Optical flow (non-rigid mapping)

Vector field for each pixel defines mapping



Problem: Without constraint arbitrary matches are possible

Grid-based elastic matching

(1) Mapping function is a grid of displacement vectors
- Lower resolution than the image
- Displacement vectors of pixels are interpolated (linear or spline)
- Penalty for relative displacement compared to neighbours

(2) Iterative refinement
- Starting with rigid registration (= one vector)
- Subdividing grid
- Initializing subgrid with parent vectors and optimizing (with lower displacement)

(3) Problem: background areas
- Elastic matching can produce arbitrary results in areas without signals
- No clear optimization direction (noise maps to noise)

Point matching

(1) Registration between arbitrary point sets
- Each point has a coordinate
- Mapping function is assignment
- Outliers may be unassigned

(2) Grid constrains mapping
- For non-rigid mappings

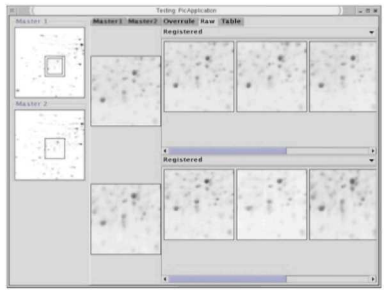
(3) Point attributes can be evaluated
- As a correlation function

Example: Proteomics data analysis

2D electrophoresis
- Quantitative protein detection as spots on a gel
- E.g. looking for cancer

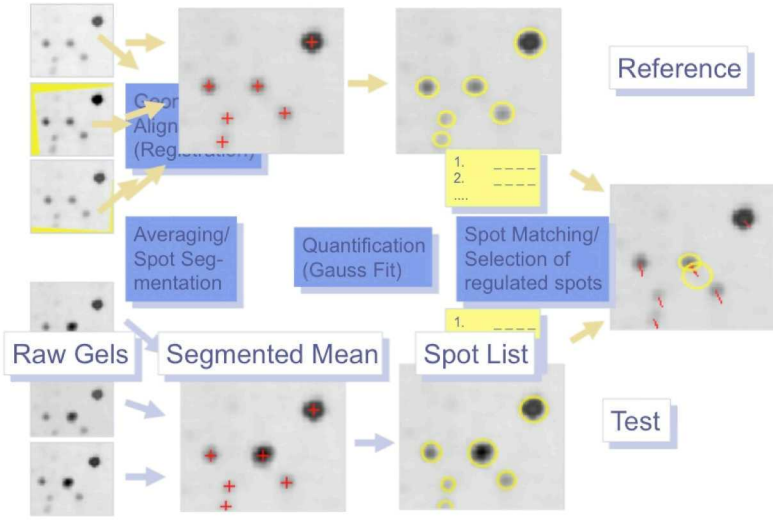
AgeLab Pharma

Aventis Research & Technologies



Intelligent data analysis
- Identification und quantification through statistical image processing
- Matching of different experiments (identity of spots unknown!)

Visual evaluation
- Presentation on natural background
- Pattern analysis, quality control and communication by the users



Registration Problems

(1) Assume identical samples and processing
- Fluctuations in the electrophoretic separation
- Staining variations
- Gel distortions

(2) Registration
- Based on mutual information (robust against staining and scanning differences)
- Allows non-rigid deformation, constrained mesh of control points

(3) Problems
- Some proteins vary their spot positions more than their neighbours
- Average of intensities smears spots

Problems of Quantification

(1) Multiple spots
- As many as 30% of the spots contain more than one protein
- Center may coincide or vary slightly (complex mixture of gaussians)

(2) Ridges
- Proteins that do not completely denaturate smear over a broad range
- Contribute to the background
- Use a local background for fitting, but sum based on global background

(3) Saturation
- Staining may saturate at high intensities
- Ignore those pixels in fitting

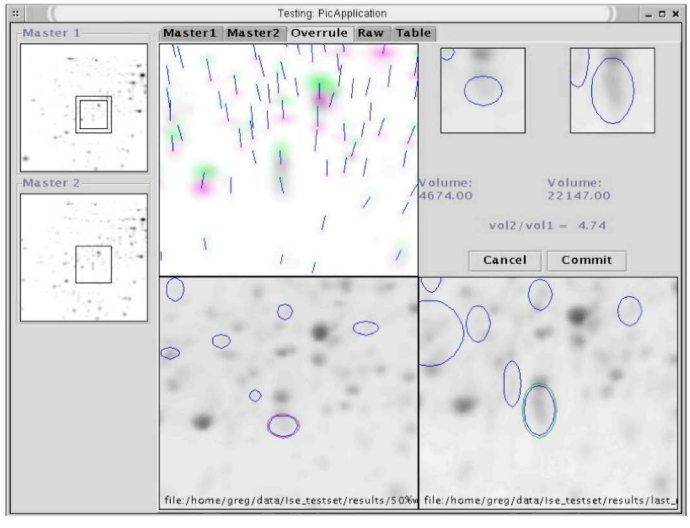
Matching different samples

(1) High variation
- Different runs, different processes
- Different biological conditions (protein concentrations may rise in minutes)

(2) Spot assignment solely based on position
- Calculation of an (affine/non-affine) mapping function and of a spot assignment by deterministic annealing

(3) Problems
- Multiple spots in one sample may separate in another
- No gold standard (without exhaustive identification via MS)

Spot assignment



3.4

Data analysis

Questions for microarray analysis

(1) Which are the different patterns of gene expression?
- Which genes closely share a pattern with gene X?

(2) What uncharacterized genes have similar expression patterns to well-characterized ones?
- What category of function might gene X belong to?

(3) Are there subtypes of disease X discernible by tissue gene expression?
- Which genes best differentiate between different tissue classes?

Data analysis ingredients

(1) Normalization
- Making values comparable across experiments

(2) Data mining
- Significant fold change
- Principal component analysis
- Clustering (unsupervised/supervised)

(3) Visualization
- Quality control
- Data visualization
- Visual data mining

Normalization

(1) Total intensity
- Assume that total DNA / protein content is always the same

(2) Regression
- Between two samples
- Linear / non-linear

(3) Set of known markers
- Housekeeping genes
- Added known RNA (corrects for purification efficiency)

→ Logarithmic expression ratio/level

(Absolute expression is restricted to positive - non-gaussian, logarithmic is assumed to be of gaussian distribution)

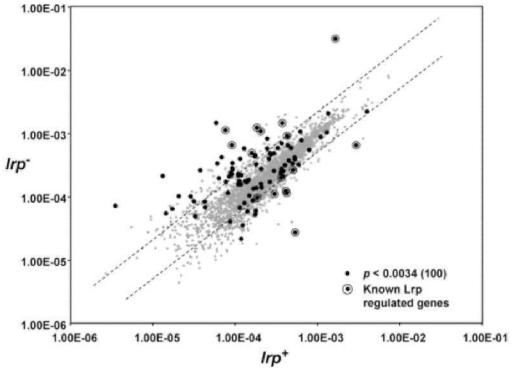
(N-)Fold analysis

(1) Comparison of two states
- Taking ratio between expression measurement
- Which ratios are significant (1.7, 2, 3, 10)?

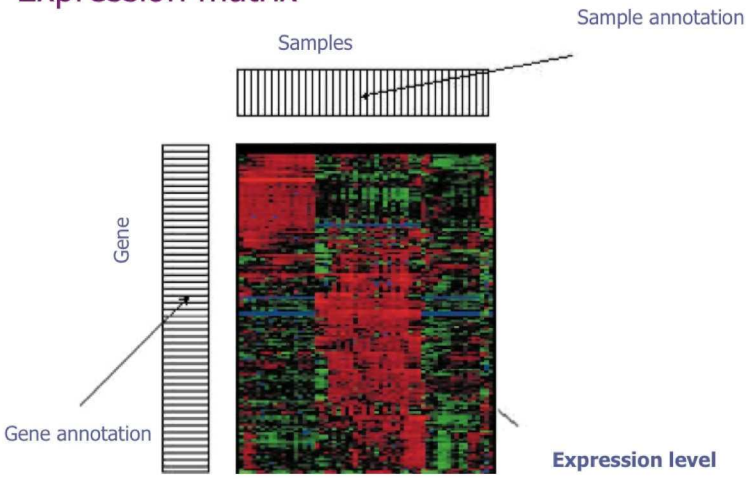
(2) Error sources
- Biological errors (fluctuation through regulation): often large, increase with expression levels, can be estimated by repeated measurements
- Technical errors (background noise): often smaller, but decrase with expression levels (signal-to-noise)

(3) Statistical estimates
- Depend on error model (of the particular array technology)
- t-test with desired level of confidence

Comparison plot of two states



Expression matrix



Cluster analysis

(1) Interpret differences in expression
- Finding groups which share a similar pattern of expression
- E.g. same reaction to different expression conditions (state, treatment)

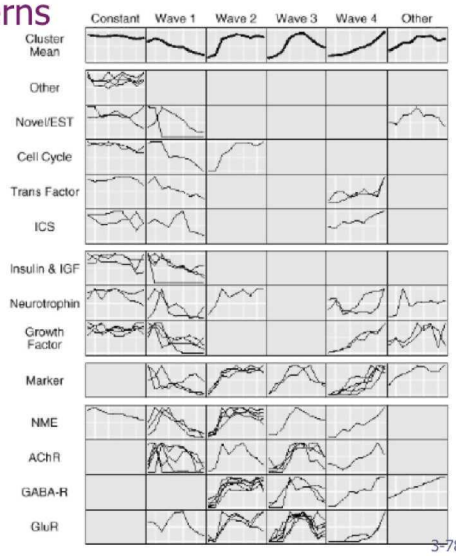
(2) Cluster analysis is a probabilistic process
- Clusters depend on a measure of distance/difference
- Clustering methods are somewhat arbitrary as they represent basically no prior knowledge

(3) Better analysis using error models
- Incorporating estimates of experimental error

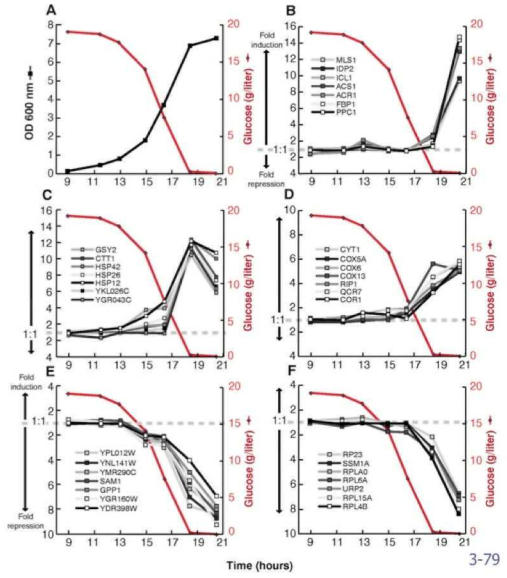
(4) New methods emerging
- Incorporating models of the biological behaviour studied

Expression patterns

Comparing different „waveforms" with different fuctional gene classes

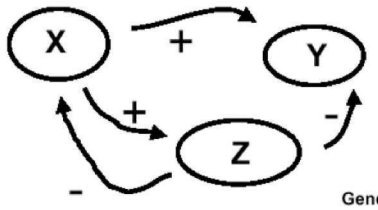


Diauxic shift



3-79

Regulatory networks



Gene Functions: X, Y, Z
Pathways: X-Y, X-Z-Y, X-Z-X
Networks: XYZ

3-80

Distance measures for clustering

(1) Each gene in the expression matrix is an n-dimensional vector
- Can be ordered (time series)
- Can be independent coordinates
- Can be arbitrary attributes

(2) Geometric distance
- Euclidian distance (dimensions must be normalized)

(3) Similarity
- Correlation
- Mutual information (suitable also for inverse regulation)

K-means clustering

(1) Given number of clusters k
- With coordinate (centroid)
- With set of assigned genes

(2) Initialize randomly to genes
- Select k genes randomly and assign each coordinate to one cluster

(3) While (clusters have changed)
- Assign each gene to its nearest centroid
- Calculate new centroids as the mean of the coordinates of the assigned genes

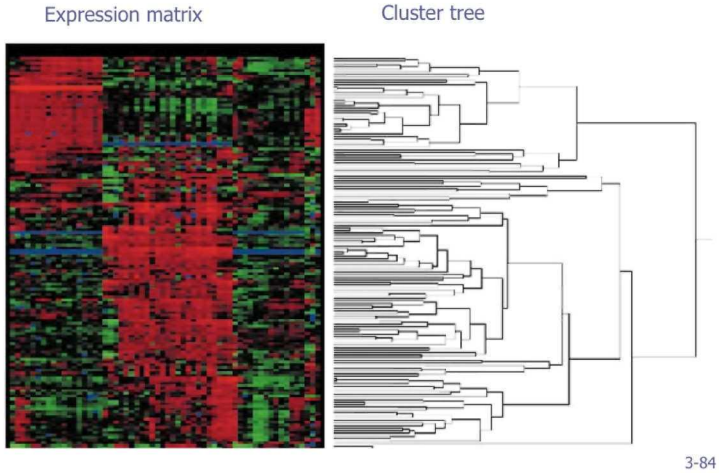
Dendrograms

(1) Built like a phylogenetic tree
- UPGMA clustering of genes

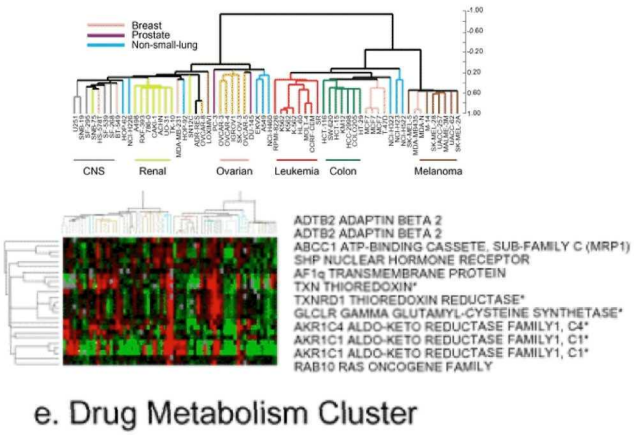
(2) Sort genes hierarchically according to tree
- Can be shown together with sorted expression matrix

(3) 2-dimensional dendrograms
- Sort samples according to clustering on attributes

Example clustering



2-dimensional dendrograms



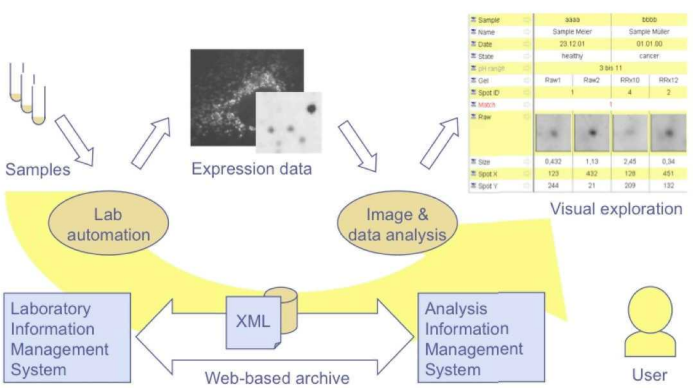
e. Drug Metabolism Cluster

From: Ross et al. Nature Genetics 24(3):227-35, 2000

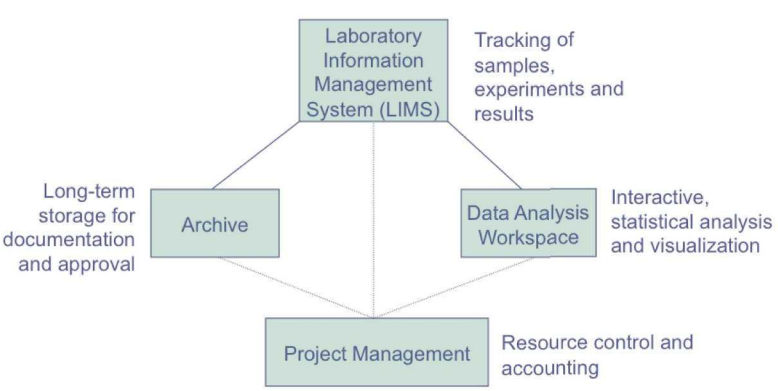
3.5

Data management

Discovery knowledge management



Data management overview



Raw Data Archiving

(1) Instrument data need to be archived
- FDA approval etc.
- Proof of authenticity needed (sigital signatures?)
- Long-term archive
  (ASCII formats like XML, or Adobe PDF)

(2) Laboratory information management system
- Track samples and experiments
- Link to archive
- Link to data analysis/project management

(3) Both should be based on a database service
- Commercial products available

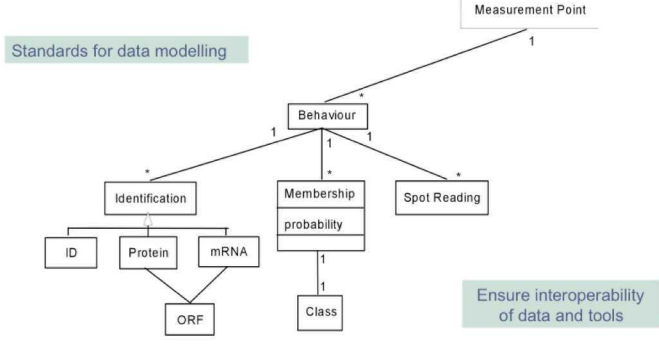
Data Design for Individual Studies

(1) Depends on experiment
- Screening (fixed protocol)
- Variation experiment (experiment-specific input parameters)

(2) Varies often
- New experimental designs
- New devices / protocols

(3) Data model needs to be planned during experiment design

Object Models



Databases vs. XML Integration

(1) Database
- Single schema, large amount of similarly-structured data
- Queries to find particular records (but overview difficult)
- Easy to update selected records
- Often monolithic and expensive

(2) Single files (Tabular ASCII, Excel, XML)
- More complex structure possible, medium amount of data
- Queries often clumsy (but browsing and overview possible)
- Selected update needs rewriting whole file
- Individual and flexible

XML Example

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE BinaryComparator PUBLIC "-//AVENTIS//DTD BinaryComparator 1.0//EN"
  "BinaryComparator.dtd">
<BinaryComparator
  numRefSpots="507"
  numTestSpots="487"
  refBioFilename="ctrl-output-bio50withBest.xml"
  refTechFilename="ctrl-output-tech50withBest.xml"
  testBioFilename="outBioIseRef50withBest.xml"
  testTechFilename="outTechIseRef50withBest.xml"
>

....

</BinaryComparator>
```

```
<Comparator
  refLabel="2"
  testLabel="2"
  refCentX="385"
  refCentY="63"
/>
<Comparator
  refLabel="3"
  testLabel="1"
  refCentX="364"
  refCentY="63"
/>
...
```

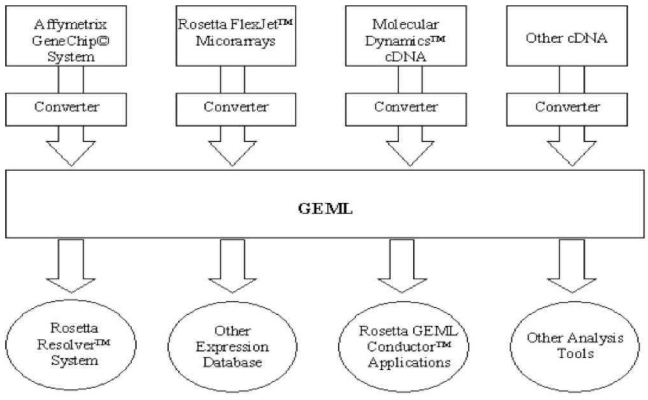

Modeling and Processing with XML

(1) XML files may contain sections of different type and structure
- Nested
- Defined in a Document Type Definition (DTD)

(2) Processing can be defined for individual sections
- Component-oriented software development

(3) Sections can be extended later
- Similar to subclassing
- Old code ignores new data, new code can handle both

Gene Expression Markup Language



GEML document types

(1) Pattern files
- Project
- Chip layout
- Probes

(3) Profile files
- Signal
- Background
- Channel information (optical)
- Log ratio

XML for expression data management

GENE				
GENE_ID	*CONTIG_ID*	*CONTIG_START*	*CONTIG_END*	*CONTIG_STRAND*
GB2VN	NT_0106058.3	2354807	2360778	Complement
GB2VN32	NT_0106051.3	2308745	2321072	Complement

```
<gene_features>
   <gene_id>GBVN32</gene_id>
   <contig_id>NT_010651</contig_id>
   <contig_start>2354807</contig_start>
   <contig_end>2360778</contig_end>
   <contig_strand>Complement</contig_strand>
</gene_features>
```

Part 4

Networks and Systems

The cell as a system

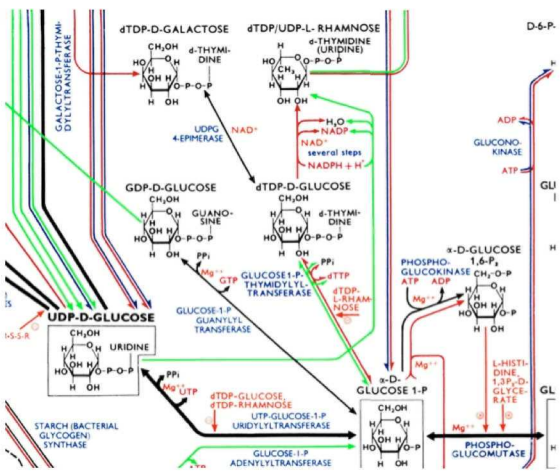
(1) Many different processes going on in parallel

- Gene expression
- Protein synthesis
- Transport
- Metabolism
- Signalling
- Regulation

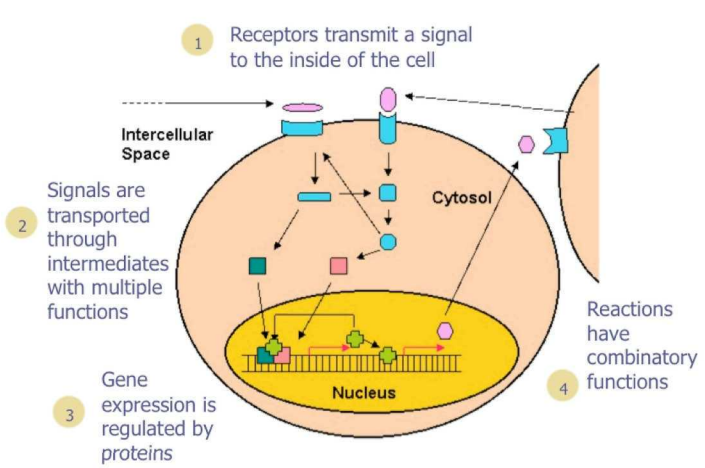
(2) Can be modelled as networks of reactions

- E.g., gene expression is a reaction with the promotors and inhibitors as inbut and the mRNA as output
- Is more or less active (number of copies, duration)

Biochemical pathways



Signal transduction in the cell



4.1

Biological networks

5

Biochemical networks

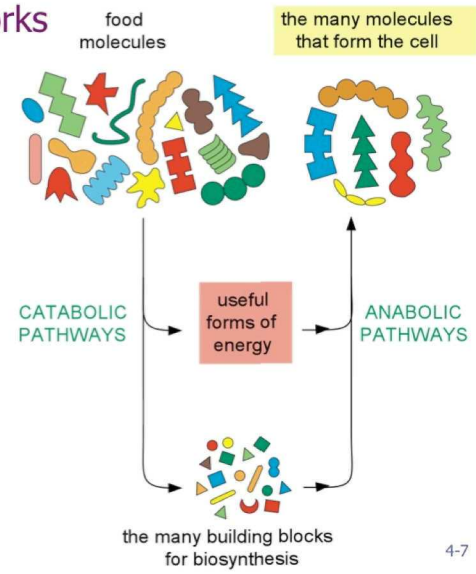
(1) Metabolic networks
- Biochemical cell functions

(2) Gene regulation networks
- Controlling cell function depending on state and external influences

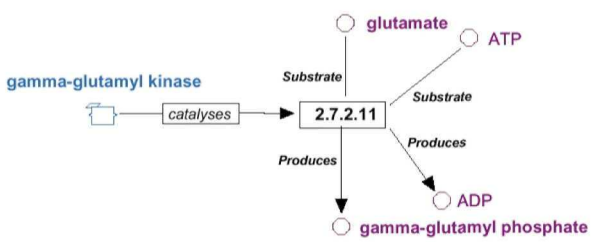
(3) Signal transduction networks
- Responding to intracellular and extracellular signals

(4) Transport
- Between different compartments of the cell and to the outside

Metabolic networks

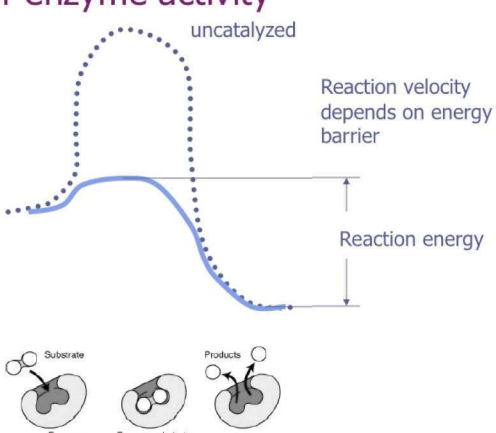


The indidvidual metabolic reaction



Substrate = input
Product = output
Enzyme = catalyzing protein

Mechanism of enzyme activity

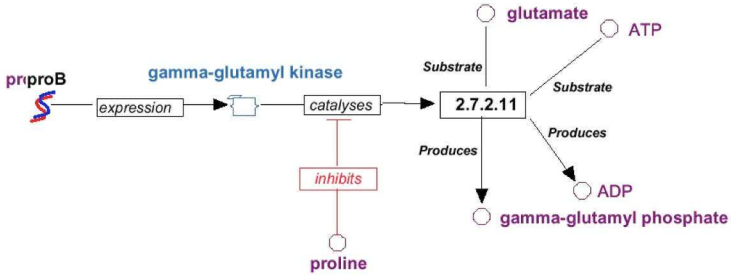


$$K = \frac{[C]\,[D]}{[A]\,[B]}$$

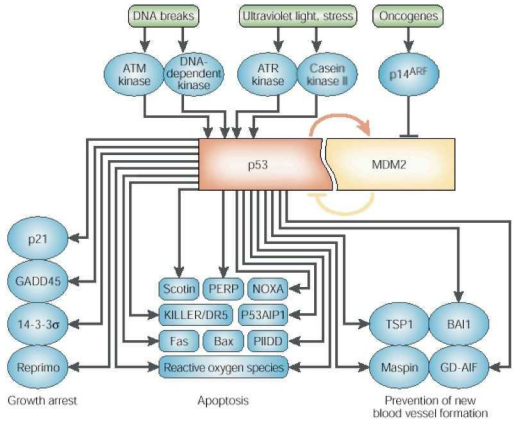
Equilibrium constant

$$A + B \longleftrightarrow C + D$$

Metabolic step

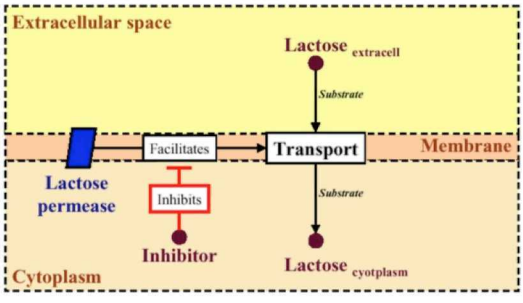


Genetic network



From Hasty et al., Nature Genetics 2:268-79, 2001.

Transport



4.2

Network databases

Metabolic pathway databases

Database	Website	Contents
BRENDA	http://www.brenda.uni-koeln.de/	Comprehensive non-restricted enzyme information
EcoCyc	http://ecocyc.org/ecocyc/ecocyc.html?	Genome and biochemical information on *Escherichia coli*
EMP project	http://emp.mcs.anl.gov/	Enzyme purification and properties as given in single references
KEGG	http://www.genome.ad.jp/kegg/kegg2.html	Metabolic and regulatory data with emphasis on sequenced genes
MEROPS	http://merops.iapc.bbsrc.ac.uk/	Information on proteases
MetaCyc	http://ecocyc.org/ecocyc/metacyc.html?	Pathways, reaction and enzymes of microorganisms
PKR	http://www.sdsc.edu/kinases/	Information on protein kinases
UMBBD	http://umbbd.ahc.umn.edu/	Biocatalytic reactions and degradation of xenobiotics

KEGG

(1) http://www.genome.ad.jp/kegg/ Institute for Chemical Research, Kyoto University.

(2) Repository of metabolic pathways for organisms whose genome is completely sequenced. Also regulatory information.

(3) For many of these organisms, the body of experimental data is very restricted. Protein function inferred from sequence similarity with proteins characterised experimentally in other organisms.

(4) Pathways represented as diagrams, manually created & stored as static gif files.

(5) Upon selection of an organism, the reactions for which an enzyme is known in that organism are highlighted in colour in the generic pathway diagrams.

EcoCyc

(1) http://ecocyc.panbio.com/ecocyc/

(2) Originally a database on metabolic pathways in Escherichia coli

(3) Currently being extended to other microbial organisms. MetaCyc (no genomic data)

(4) Pathway diagrams have been generated with the aid of a graph layout algorithm, & stored as static images for web browsing.

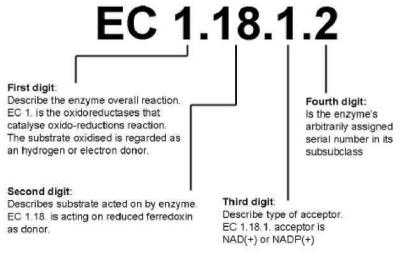
(5) Based on published experimental data, and unlike KEGG, also includes information about genes that have not yet been cloned but whose function has been characterised by genetic and biochemical approaches.

BRENDA

Information field	Total entries	Information field	Total entries
Enzyme nomenclature		**Functional parameters**	
EC number	3869	K_m value	28 134
Systematic name	3182	Specific activity	11 787
Synonyms	17 707	pH optimum	14 037
CAS registry number	3552	pH range	3929
Reaction	3518	Temperature optimum	6147
Reaction type	4123	Temperature range	908
Enzyme structure		**Molecular properties**	
Molecular weight	12 329	pH stability	2931
Subunits	7416	Temperature stability	6825
Sequence links	33 099	General stability	5398
Posttranslational modification	1112	Organic-solvent stability	311
Crystallization	1003	Oxidation stability	349
3D-structure, specific PDB links	6142	Storage stability	6505
		Purification	11 176
Enzyme- ligand interactions		Cloned	2015
Substrates–products	47 630	Engineering	797
Natural substrate	7668	Renatured	199
Cofactor	6217	Application	338
Activating compound	6217		
Metals–ions	13 173	**Organism-related information**	
Inhibitors	56 336	Organism	40 027
		Source tissue, organ	19347
Bibliographical data		Localization	7935
References	46 305		

EC Classification

- Classified according to Enzyme Nomenclature (IUBMB)
- Six major biochemical reactions



TRANSPATH®

(1) Database on molecular pathways and cellular network modelling

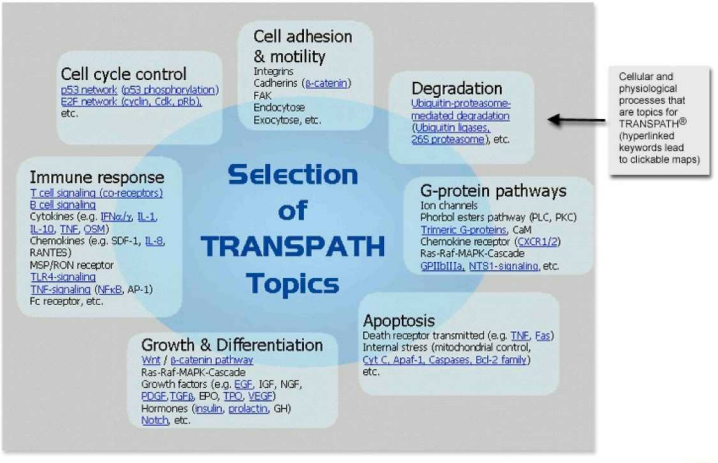
- all topics of molecular interactions within the cytoplasm of a cell
- including on-the-fly generation of pathways from extra-cellular agents cascading all the way down
- for example, to the affected transcription factor and the gene triggered

(2) Elements (molecules) of relevant signal transduction pathways

- ligands, receptors, enzymes and transcription factors,
- molecular interactions between single molecules.

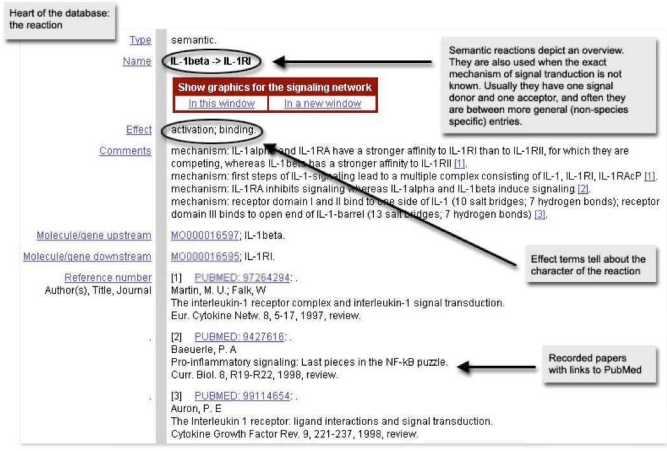
http://www.biobase.de/pages/products/transpath.html

TRANSPATH topics

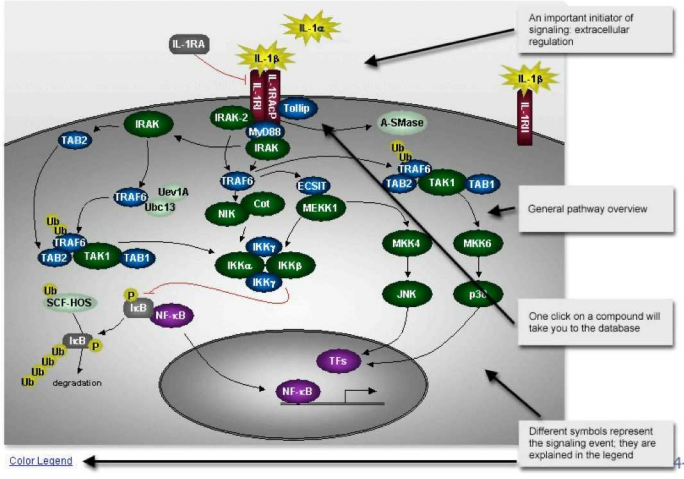


Molecule entry

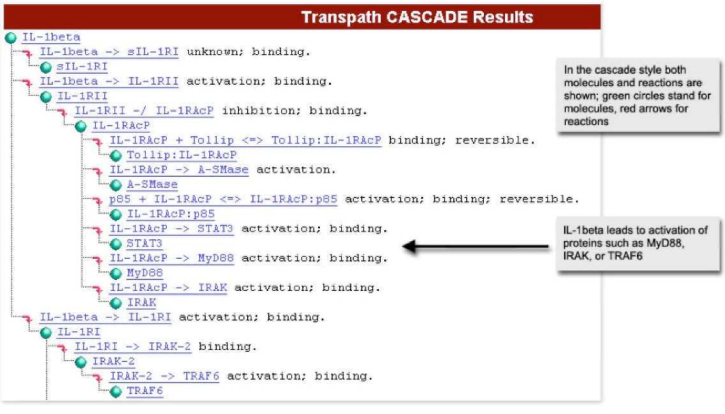
Reaction entry



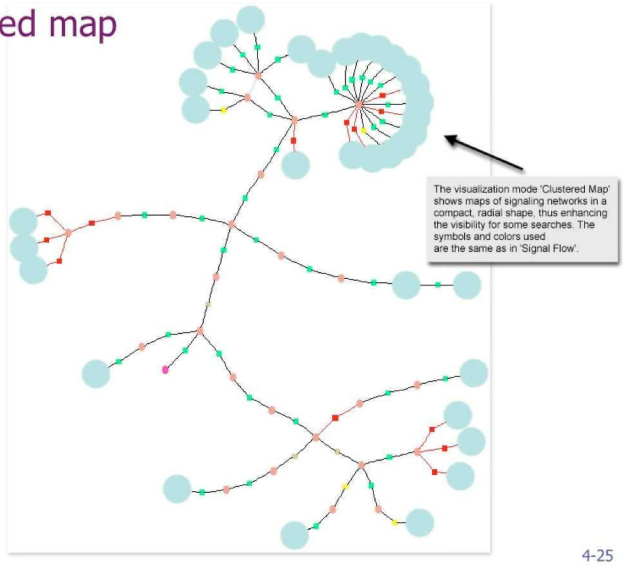
Clickable map (hand-drawn)



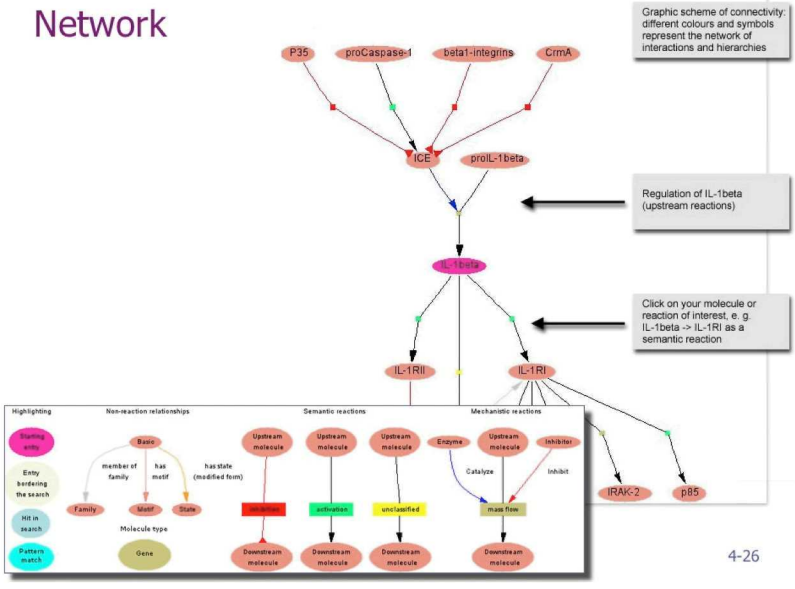
Cascade style pathway visualization



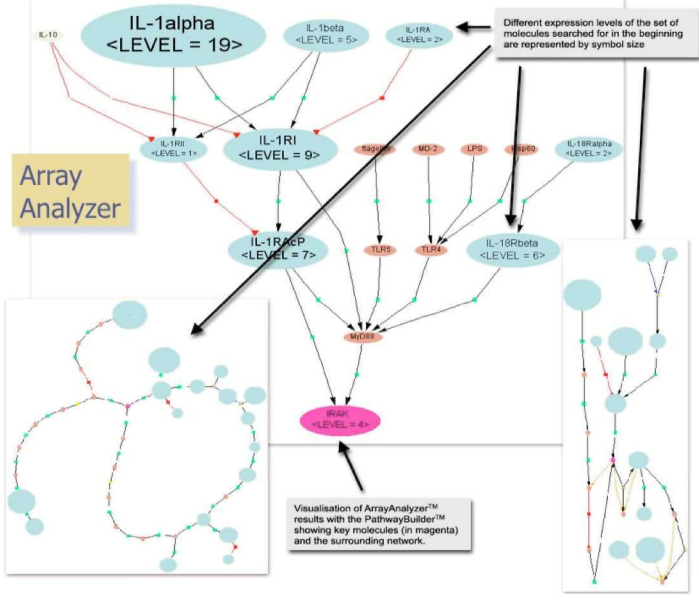
Clustered map



Network



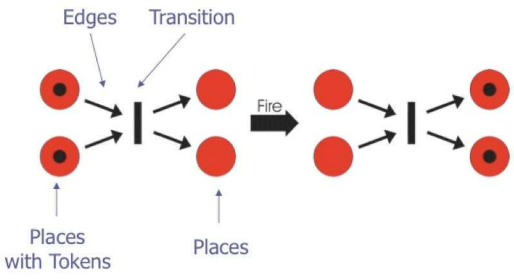
Array Analyzer



4.3

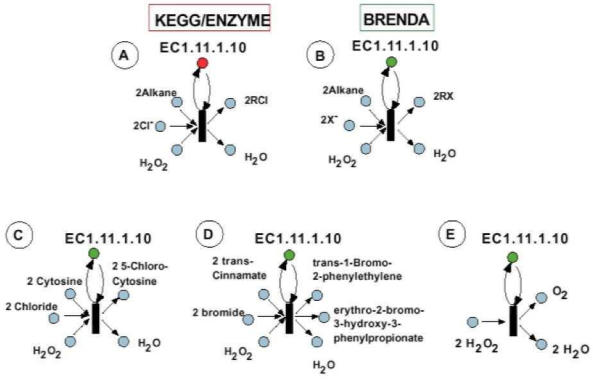
Network analysis

Petri Net Firing Rule



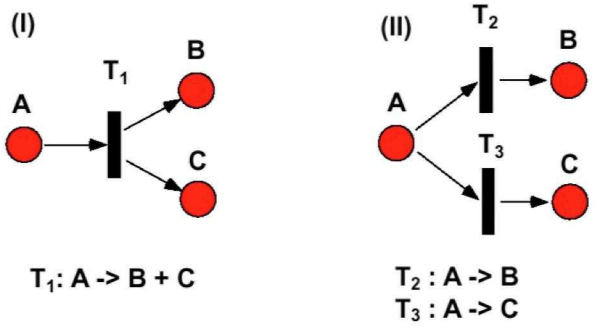
$N = (P, T, E)$ $E \subseteq (P \times T) \cup (T \times P)$

Reactions as Petri Nets



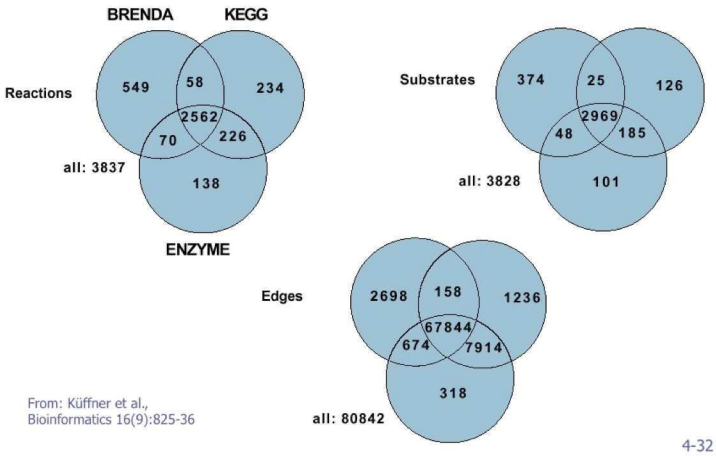
From: Küffner et al., Bioinformatics 16(9):825-36

Conflicting reactions with competing input



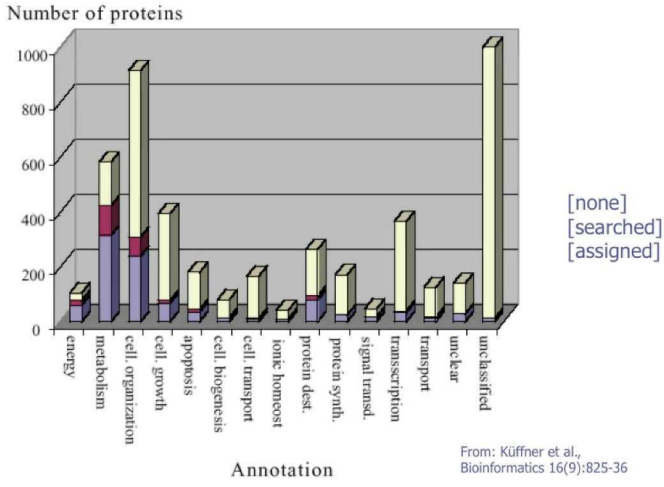
Firing rule distinguishes cases compared to simple graphs

Comparison of database content



From: Küffner et al., Bioinformatics 16(9):825-36

Yeast genes covered by EC numbers



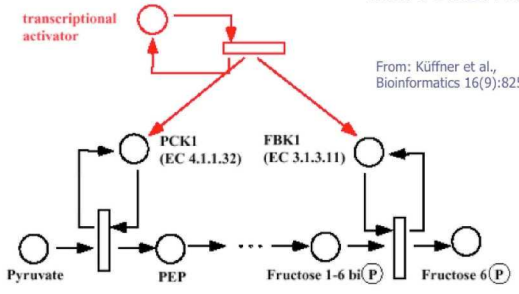
[none]
[searched]
[assigned]

From: Küffner et al., Bioinformatics 16(9):825-36

Extension through other DBs

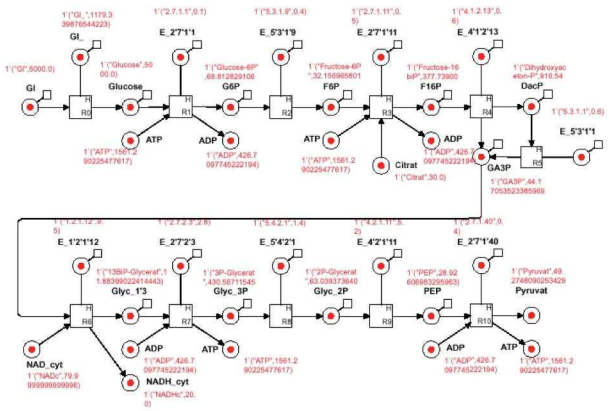
```
ID   CAT8_YEAST      STANDARD;      PRT;  1433 AA.
AC   P39113;
DE   REGULATORY PROTEIN CAT8.
GN   CAT8 OR MSP8 OR YMR280C OR YM8021.06C.
OS   SACCHAROMYCES CEREVISIAE (BAKER'S YEAST).
OC   EUKARYOTA; FUNGI; ASCOMYCOTINA; HEMIASCOMYCETES.
...
CC   -!- FUNCTION: ACTIVATOR OF THE GLUCONEOGENIC
CC       ENZYMES FBP1 AND PCK1 GENES.
CC   -!- SUBCELLULAR LOCATION: NUCLEAR.
CC   -!- PTM: COULD BE THE TARGET OF THE SNF1/CAT1 -
```

Extension through regulatory information from a SwissProt entry



From: Küffner et al., Bioinformatics 16(9):825-36

Textbook pathway of glycolysis



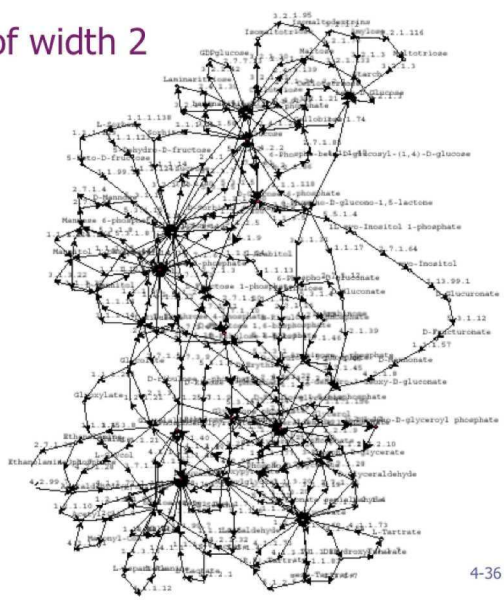
From: Küffner et al., Bioinformatics 16(9):825-36

All pathways of width 2

541 pathways

(80000 paths with 800 enzymes without restriction and length <9)

Shows biological robustness (lots of alternate paths)

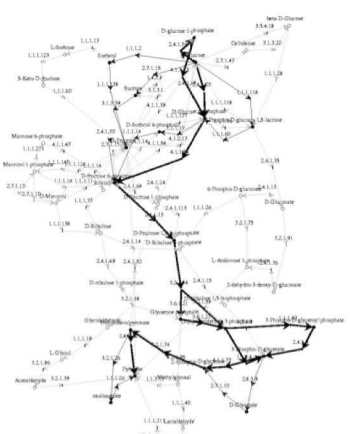


From: Küffner et al., Bioinformatics 16(9):825-36

Differential metabolic display

Display valid pathways for certain organisms/states (here yeast and mycoplasma genitalium)

Thick = both
Thin = yeast only
Light gray = neither



From: Küffner et al., Bioinformatics 16(9):825-36

Pathway scoring

(1) Selection of interesting pathways
- Closed set
- Particular organism

(2) Gene expression data
- Related to enzymes in the pathway

(3) Scoring
- Highlighting relevant genes
  gene score is calculated on gene expression data
- Highlighting relevant paths
  path score is calculated over whole path

Conspicousness score

(1) Double measurement of expression values
- Gives mean and standard deviation of measurement error (null model)
- Mean is near zero

(2) Score evaluates distance from mean in normal distribution

$$P_t^0(g) = 2\Phi\left(-\left|\frac{m_{t,g} - 0}{s_{err}}\right|\right) \qquad score(g) = -\log P_t^0(g)$$

(3) Score of time sequence is sum of scores
- Independence (multiplication of probabilities)

$$score(g) = \frac{1}{|T| - 1} \sum_{t \in T - \{t_0\}} score_t(g)$$

Pathway score

(1) Average conspicousness over path

$$score(p) = \frac{1}{|p|} \sum_{g \in p} score(g)$$

Scoring synchrony of expression

(1) Gene scoring

- Correlation coefficient to all other genes in the pathway

$$score_p(g) = \frac{1}{|p_g|} \sum_{h \in p_g} cc(g,h)$$

- Excluding the gene itself if it is member of the pathway

(2) Pathway scoring

- Average of all gene scores over the pathway

$$score(p) = \frac{1}{|p|} \sum_{g \in p} score_p(g)$$

Combined scores

(1) Synchrony scoring has advantages over clustering

- Synchrony over pathway is effective even if genes are in different clusters
- Arbitrary synchrony to other genes is disregarded outside the pathways
- But gives high scores for synchronous but inconspicous results

(2) Modified correlation coefficient

- With standard error in denominator instead of individual variance

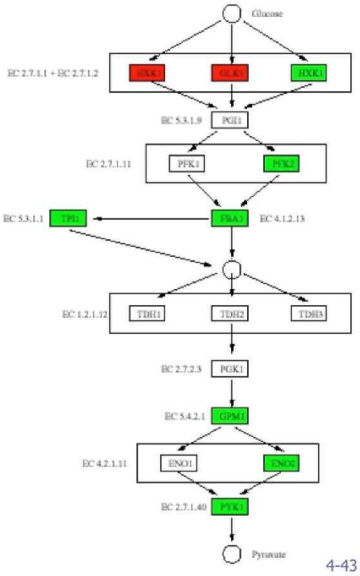
(3) Score of random pathway is 1

Example

Different ORFs for each EC number

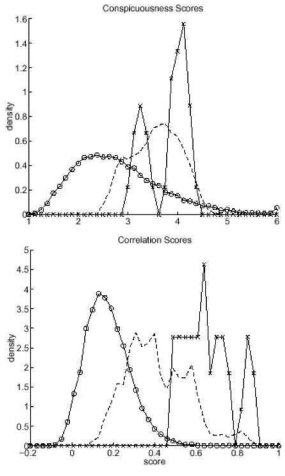
36 pathways altogether

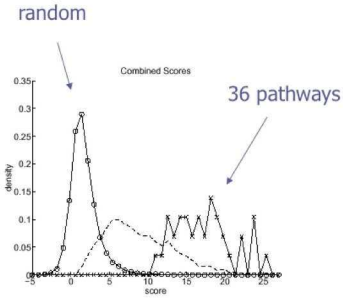
Tested with all three scoring functions



From: Zien et al., ISMB2000

Histograms





From: Zien et al., ISMB2000

Results

Colours mean up- or downregulation of individual genes

HXK2 is most significant gene score

TDL1, TDL2, TDL3 make no difference

Genes	HXK1					TDH1				
	GLK1		PFK1			TDH2			ENO1	
Scores	HXK2	PGI1	PFK2	FBA1	TPI1	TDH3	PGK1	GPM1	ENO2	PYK1
0,869	YGL253W	YBR196C	YMR205C	YKL060C	YDR050C	YJR009C	YCR012W	YKL152C	YKR174W	YAL038W
0,867	YGL253W	YBR196C	YMR205C	YKL060C	YDR050C	YGR192C	YCR012W	YKL152C	YKR174W	YAL038W
0,860	YGL253W	YBR196C	YMR205C	YKL060C	YDR050C	YJL052W	YCR012W	YKL152C	YKR174W	YAL038W
0,838	YGL253W	YBR196C	YGR240C	YKL060C	YDR050C	YJR009C	YCR012W	YKL152C	YKR174W	YAL038W
0,837	YGL253W	YBR196C	YGR240C	YKL060C	YDR050C	YGR192C	YCR012W	YKL152C	YKR174W	YAL038W
0,810	YGL253W	YBR196C	YGR240C	YKL060C	YDR050C	YJL052W	YCR012W	YKL152C	YKR174W	YAL038W
0,748	YGL253W	YBR196C	YMR205C	YKL060C	YDR050C	YGR192C	YCR012W	YKL152C	YOR254W	YAL038W
0,748	YGL253W	YBR196C	YMR205C	YKL060C	YDR050C	YJR009C	YCR012W	YKL152C	YOR254W	YAL038W
0,742	YGL253W	YBR196C	YMR205C	YKL060C	YDR050C	YJL052W	YCR012W	YKL152C	YOR254W	YAL038W
0,721	YGL253W	YBR196C	YGR240C	YKL060C	YDR050C	YGR192C	YCR012W	YKL152C	YOR254W	YAL038W
0,721	YGL253W	YBR196C	YGR240C	YKL060C	YDR050C	YJR009C	YCR012W	YKL152C	YOR254W	YAL038W
0,715	YGL253W	YBR196C	YGR240C	YKL060C	YDR050C	YJL052W	YCR012W	YKL152C	YOR254W	YAL038W
0,690	YFR053C	YBR196C	YMR205C	YKL060C	YDR050C	YJR009C	YCR012W	YKL152C	YKR174W	YAL038W
0,687	YFR053C	YBR196C	YMR205C	YKL060C	YDR050C	YGR192C	YCR012W	YKL152C	YKR174W	YAL038W
0,685	YFR053C	YBR196C	YMR205C	YKL060C	YDR050C	YJL052W	YCR012W	YKL152C	YKR174W	YAL038W
0,655	YFR053C	YBR196C	YGR240C	YKL060C	YDR050C	YJR009C	YCR012W	YKL152C	YKR174W	YAL038W
0,653	YFR053C	YBR196C	YGR240C	YKL060C	YDR050C	YGR192C	YCR012W	YKL152C	YKR174W	YAL038W
0,650	YFR053C	YBR196C	YGR240C	YKL060C	YDR050C	YJL052W	YCR012W	YKL152C	YKR174W	YAL038W
0,638	YCL040W	YBR196C	YMR205C	YKL060C	YDR050C	YJR009C	YCR012W	YKL152C	YKR174W	YAL038W
0,637	YCL040W	YBR196C	YMR205C	YKL060C	YDR050C	YGR192C	YCR012W	YKL152C	YKR174W	YAL038W
0,634	YCL040W	YBR196C	YMR205C	YKL060C	YDR050C	YJL052W	YCR012W	YKL152C	YKR174W	YAL038W
0,604	YCL040W	YBR196C	YGR240C	YKL060C	YDR050C	YJR009C	YCR012W	YKL152C	YKR174W	YAL038W
0,604	YCL040W	YBR196C	YGR240C	YKL060C	YDR050C	YGR192C	YCR012W	YKL152C	YKR174W	YAL038W
0,600	YCL040W	YBR196C	YGR240C	YKL060C	YDR050C	YJL052W	YCR012W	YKL152C	YKR174W	YAL038W
0,583	YFR053C	YBR196C	YMR205C	YKL060C	YDR050C	YJR009C	YCR012W	YKL152C	YOR254W	YAL038W
0,582	YFR053C	YBR196C	YMR205C	YKL060C	YDR050C	YGR192C	YCR012W	YKL152C	YOR254W	YAL038W
0,580	YFR053C	YBR196C	YMR205C	YKL060C	YDR050C	YJL052W	YCR012W	YKL152C	YOR254W	YAL038W
0,552	YFR053C	YBR196C	YGR240C	YKL060C	YDR050C	YJR009C	YCR012W	YKL152C	YOR254W	YAL038W
0,551	YFR053C	YBR196C	YGR240C	YKL060C	YDR050C	YGR192C	YCR012W	YKL152C	YOR254W	YAL038W
0,549	YFR053C	YBR196C	YGR240C	YKL060C	YDR050C	YJL052W	YCR012W	YKL152C	YOR254W	YAL038W
0,528	YCL040W	YBR196C	YMR205C	YKL060C	YDR050C	YGR192C	YCR012W	YKL152C	YOR254W	YAL038W
0,527	YCL040W	YBR196C	YMR205C	YKL060C	YDR050C	YJR009C	YCR012W	YKL152C	YOR254W	YAL038W
0,525	YCL040W	YBR196C	YMR205C	YKL060C	YDR050C	YJL052W	YCR012W	YKL152C	YOR254W	YAL038W
0,499	YCL040W	YBR196C	YGR240C	YKL060C	YDR050C	YGR192C	YCR012W	YKL152C	YOR254W	YAL038W
0,496	YCL040W	YBR196C	YGR240C	YKL060C	YDR050C	YJR009C	YCR012W	YKL152C	YOR254W	YAL038W
0,495	YCL040W	YBR196C	YGR240C	YKL060C	YDR050C	YJL052W	YCR012W	YKL152C	YOR254W	YAL038W

From: Zien et al., ISMB2000

4.4

Simulation

Simulation of functional networks

(1) Given comprehensive network of reactions
- Reaction constants: equilibrium, speed etc.

(2) Conditions
- Number of molecules/concentration
- State of molecules (docked, phosphorylated, etc.)
- Location (within compartments)

(3) Simulation over time
- Given initial conditions
- Observe simulated behaviour over time
- Watch individual molecules
- Observe temporal functions (regulation, etc.)
- Observe global patterns

Simulation methods

(1) System dynamics
- Formulation as differential equations
- Change rate is a function of the current state
- Iterative solution
- Can capture reaction kinetics, transport by diffusion

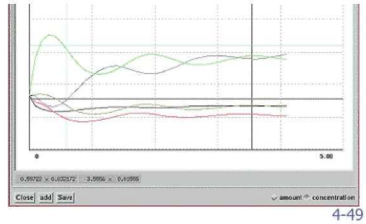
(2) Stochastic
- Formulation as probabilities of individual change
- Smaller elements (individual molecules and single steps)
- Random generation (not always reproducible!)
- Can capture all sorts of state transitions, including individual modifications and building of complexes

Example: E-Cell (Tomita)

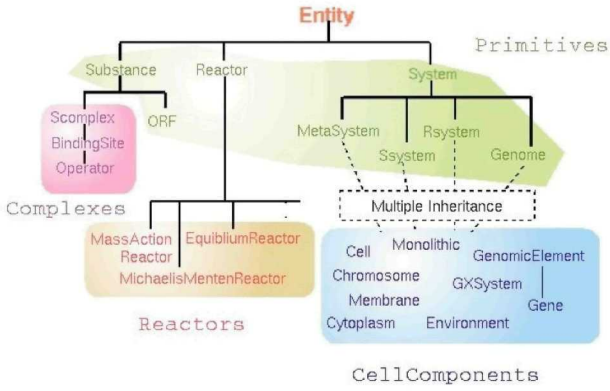
(1) Dynamic model of *Mycoplasma Genitalium*
- Small number of genes (480)

(2) Model even more reduced
- Only 127 genes
- Included: transcription, translation, energy production, phospolipid sythesis
- Excluded: proliferation (no cell cycle or DNA synthesis)

Tracing concentrations of several molecules over time



E-Cell object types



Systems Biology Markup Language (SBML)

(1) XML extension for Systems Biology
- Definition of
- Lists of Compartments, Species, Reactions
- Optional Lists of Parameters, Units and Rules

(2) Supported by
- Cellerator, Dbsolve, E-Cell, Gepasi/COPASI, Jarnac/WinScamp, ProMoT/DIVA, StochSim, Virtual Cell

(3) Kinetic rate law equations
- Simple Michaelis-Menten, Hill kinetics with modifiers, competitive/non-competitive inhibition, allosteric inhibition...
- SBML lists 34 reaction types!

Problems of kinetic laws

(1) Getting the parameters

- Measured under „artificial" conditions
- Many not known

(2) Characterizing reactions

- Exact mechanism not known (which rate law is the best approximation?)

(3) Complexity

- Considerable number of parameters for mor complex reaction types
- Example: "enzyme glutamine synthease" with eight reactants and modifiers results in a system with about 500 terms and would need more than 100 million experiments in order to determine the parameter values

S-Systems (Voit)

(1) Simpler approximation with general structure

- If we have to estimate/infer parameters and equation types anyway, we could also use some general parameterised equations with better mathematical properties
- Most differential equations can be recast into a nonlinear canonical form called S-System
- Transformation of the S-System in logarithmic coordinates allows efficient computation of higher order derivatives and the construction of a higher order Taylor solver (10-100 times faster than standard ODE solvers)

S-Systems general equation

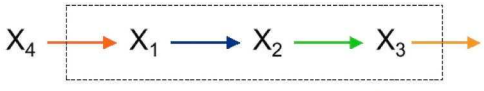
(1) S-Systems provide a single parameter for very different reaction types

- Unmodulated material flow 0.5 – 1
- Activations 0 – 0.5
- Inhibitions -0.5 – 0

$$X_i' = \alpha_i X_1^{g_{i1}} X_2^{g_{i2}} \ldots X_N^{g_{iN}} - \beta_i X_1^{f_{i1}} X_2^{f_{i2}} \ldots X_N^{f_{iN}}$$

Product Substrate

S-System for a linear pathway



$X_1' = V_1^+ - V_1^- = \alpha_1 X_4^{g_{14}} - \beta_1 X_1^{h_{11}}$

$X_2' = V_2^+ - V_2^- = \alpha_2 X_1^{g_{21}} - \beta_2 X_2^{h_{22}}$

$X_3' = V_3^+ - V_3^- = \alpha_3 X_2^{g_{32}} - \beta_3 X_3^{h_{33}}$

$X_4 = \text{const.}$

With conditions

$\alpha_2 = \beta_1,\ g_{21} = h_{11},\ \alpha_3 = \beta_2,\ g_{32} = h_{22}$

Stochastic simulation

(1) Deterministic approach relies upon assumptions that are – in particular for biological systems - not always valid:

- Infinite reaction volume
- System close to equilibrium

(2) Stochastic approach

- Is also applicable for small reaction volumes, short timescales and far from the equilibrium

Stochastic simulation

(1) Discrete state space

- Integer amounts of each species in the system

(2) Reaction probabilities instead of reaction rates

- Essentially simulation of a Markov process, i.e. one biochemical reaction per iteration

(3) Simulate the equations

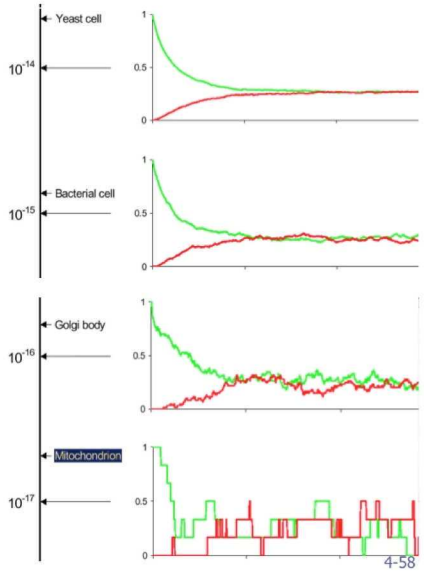
- To generate trajectories trough the state space
- Different simulation runs will produce different results

Volume effects

Simulation of a reaction with increasingly smaller reaction volumes

Final graph shows stochastic behaviour with only three molecules

[StochSim: Carl Firth]



StochSim algorithm

(1) Binary reactions of individual molecule objects
- At time t (small increments)
- Two molecules are chosen randomly
- Second choice can come from *dummy molecules* to simulate unimolecular reactions (state transitions)

(2) Choosing a probable reaction
- Scanning list of all possible reactions with their probabilities
- Choosing one randomly accounting for that probability
- Or choose none (with the remaining probability)

(3) Execute reaction
- Produce new molecules
- Or modify state of molecules

Gillespie algorithm

(1) Probability density for next reaction
- That next reaction will occur at time increment *d* and will be of type *u*
- Calculate either as „first reaction" or „next reaction"

(2) Choose next reaction
- Increment time by d
- Execute effects of reaction u

(3) „First reaction" method
- First calculate time expectation for every reaction type
- Then choose the minimum (expensive!)

(4) „Next reaction" method
- Keep „next time" for all reactions in a sorted list
- Update only those that are affected by the current reaction

Comparison StochSim - Gillespie

(1) Gillespie
- Skips always to the next „active" time interval
- Treats all molecules of a kind together
- Assumes homogeneous concentration (and unconstrained size)

(2) StochSim
- Slower (individual molecules)
- Can better account different protein states (e.g., protein with N different modifications sites needs to be regarded as 2^N different molecule classes by Gillespie)
- Can account for complex formation and geometric distribution

Summary

(1) Simulation of individual cells will be possible
- In the near future, using computer clusters and approximations of reactions
- Need to identify useful model simplifications

(2) Need to obtain biological information
- Molecules involved, possible reactions, reaction rates
- Localization in the cell

(3) Need to identify system features
- What is needed for robust regulatory functions?
- How is biological function divided into independent modules?

(4) Cognitive problem
- Concepts and notations that reduce complexity
- Need new culture techniques and languages