Modellierung und Simulation in der Biochemie

Ursula Kummer

Vorlesung 11

Methoden zur Parameterschätzung

Overview

- General Strategy
- Finding initial values
- Optimisation routines

General strategy

- Model a biochemical system
- First guess on the parameters
- Compute difference between the simulated and the experimental data
- Optimize the result by changing the parameters
- Assess the result

Review: Modeling

$glc' = v_{trans} - v_{hk}$

$g6p' = v_{hk} - v_{pgi}$

$f6p' = v_{pgi} - v_{pfk}$

$f16p' = v_{pfk} - v_{ald}$

$dhap' =$

$gap' = v$

$bpg' = v_{gpdh} - v_{pgk}$

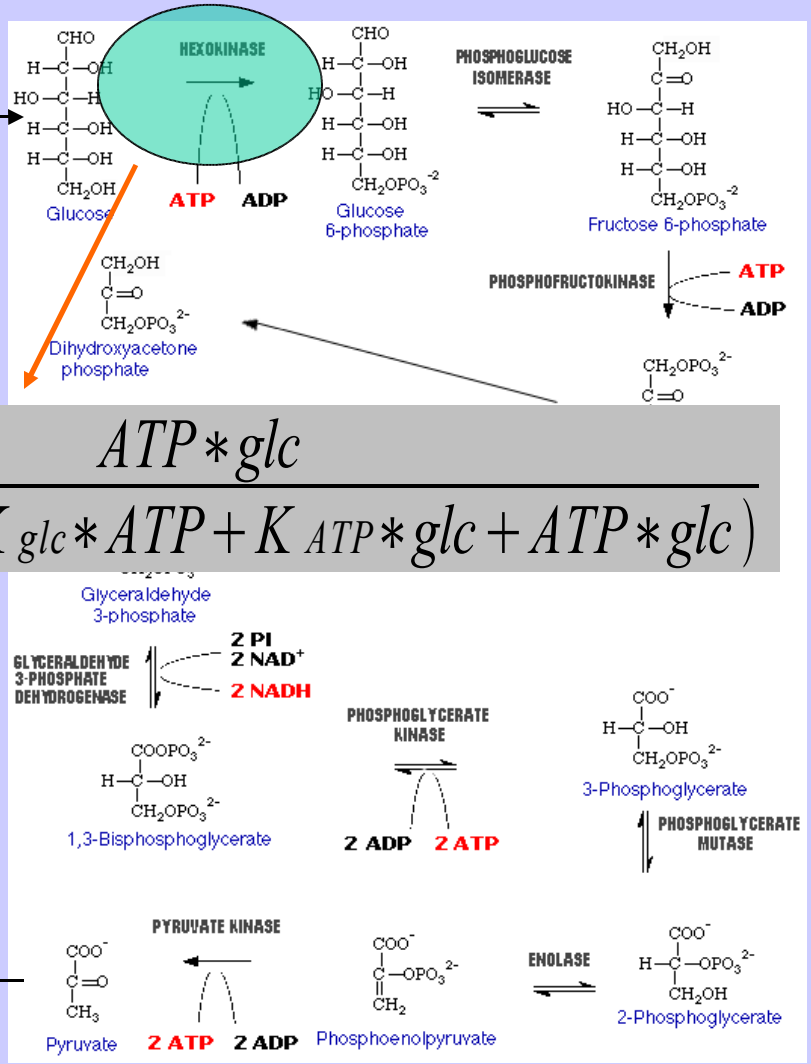
$p3g' = v_{pgk} - v_{pgm}$

$p2g' = v_{pgm} - v_{eno}$

$pp' = v_{eno} - v_{pyk}$

$py' = v_{pyk} - v_{py}$

$$v_{hk} = V_{hk} * \frac{ATP * glc}{(K_{ATP} * K_{glc} + K_{glc} * ATP + K_{ATP} * glc + ATP * glc)}$$

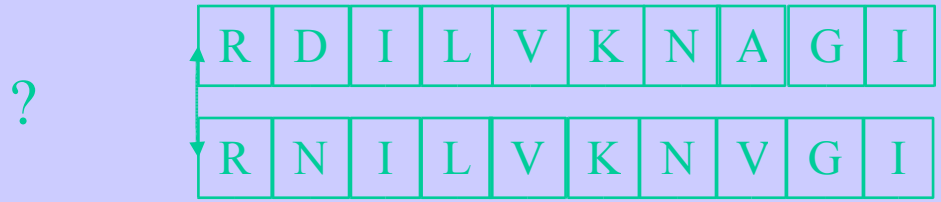


Parameters: The First Guess

- Find the parameters of the respective enzyme under similar conditions, in similar cells or organisms
- Assess the similarity of the enzymes
- Take the parameters for the most similar enzyme
- Alternatively: Study ways to compute the parameter from the structure

Sequence Comparison

- Analyses evolutionary relationship
- Similar sequence might imply similar function, structure, parameters



Identity matrix

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	1																			
R	0	1																		
N	0	0	1																	
D	0	0	0	1																
C	0	0	0	0	1															
Q	0	0	0	0	0	1														
E	0	0	0	0	0	0	1													
G	0	0	0	0	0	0	0	1												
H	0	0	0	0	0	0	0	0	1											
I	0	0	0	0	0	0	0	0	0	1										
L	0	0	0	0	0	0	0	0	0	0	1									
K	0	0	0	0	0	0	0	0	0	0	0	1								
M	0	0	0	0	0	0	0	0	0	0	0	0	1							
F	0	0	0	0	0	0	0	0	0	0	0	0	0	1						
P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1					
S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1				
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1			
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1		
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

A	B	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y	Z	
3.0	2.0	1.0	2.0	2.0	1.0	2.0	1.0	1.0	1.0	1.0	1.0	1.0	2.0	1.0	1.0	2.0	2.0	2.0	1.0	1.0	2.0	A
	3.0	1.0	3.0	2.0	1.0	2.0	2.0	2.0	2.0	1.0	1.0	3.0	1.0	2.0	1.0	2.0	2.0	2.0	0.0	2.0	2.0	B
		3.0	1.0	0.0	2.0	2.0	1.0	1.0	0.0	1.0	0.0	1.0	1.0	0.0	2.0	2.0	1.0	1.0	2.0	2.0	0.0	C
			3.0	2.0	1.0	2.0	2.0	1.0	1.0	1.0	0.0	2.0	1.0	1.0	1.0	1.0	1.0	2.0	0.0	2.0	2.0	D
				3.0	0.0	2.0	1.0	1.0	2.0	1.0	1.0	1.0	1.0	2.0	1.0	1.0	1.0	2.0	1.0	1.0	3.0	E
					3.0	1.0	1.0	2.0	0.0	2.0	1.0	1.0	1.0	0.0	1.0	2.0	1.0	2.0	1.0	2.0	0.0	F
						3.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	2.0	2.0	1.0	2.0	2.0	1.0	2.0	G
							3.0	1.0	1.0	2.0	0.0	2.0	2.0	2.0	2.0	1.0	1.0	1.0	0.0	2.0	2.0	H
								3.0	2.0	2.0	2.0	2.0	1.0	1.0	2.0	2.0	2.0	2.0	0.0	1.0	1.0	I
									3.0	1.0	2.0	2.0	1.0	2.0	2.0	1.0	2.0	1.0	1.0	1.0	2.0	K
										3.0	2.0	1.0	2.0	2.0	2.0	2.0	1.0	2.0	2.0	1.0	2.0	L
											3.0	1.0	1.0	1.0	2.0	1.0	2.0	2.0	1.0	0.0	1.0	M
												3.0	1.0	1.0	1.0	2.0	2.0	1.0	0.0	2.0	2.0	N
													3.0	2.0	2.0	2.0	2.0	1.0	1.0	1.0	2.0	P
														3.0	2.0	1.0	1.0	1.0	1.0	1.0	3.0	Q
															3.0	2.0	2.0	1.0	2.0	1.0	2.0	R
																3.0	2.0	1.0	2.0	2.0	1.0	S
																	3.0	1.0	1.0	1.0	1.0	T
																		3.0	1.0	1.0	2.0	V
																			3.0	1.0	1.0	W
																				3.0	1.0	Y
																					3.0	Z

Genetic Matrix

B = D∨ N

Z = E ∨ Q

Example

R	D	I	L	V	K	N	A	G	I
R	N	I	L	V	K	N	V	G	I

Identity Matrix : 1 + 0 + 1 + 1 + 1 + 1 + 1 + 0 + 1 + 1 = 8

Genetic Matrix: 3 + 2 + 3 + 3 + 3 + 3 + 3 + 2 + 3 + 3 = 28

PAM250: 6 + 2 + 5 + 6 + 4 + 5 + 2 + 0 + 5 + 5 = 40

BLOSUM62: 5 + 1 + 4 + 4 + 4 + 5 + 6 + 0 + 6 + 4 = 39

BLAST

- Basic Local Alignment Search Tool
- generates a list of local alignments between a query sequence and entries in a database

BLAST

- Starting point: Query-sequence, wordlegnth w, thresholds *S, T*
- Algorithm:
  - for w and a given score matrix all words are determined that result in a score > *T* when compared to the query sequence
  - The database is searched for these w-meres
  - Every hit (corresponding to a sequence in the DB) will be extended in both directions and checked, if a score > *S* results
  - Output of all local alignments resulting from this procedure

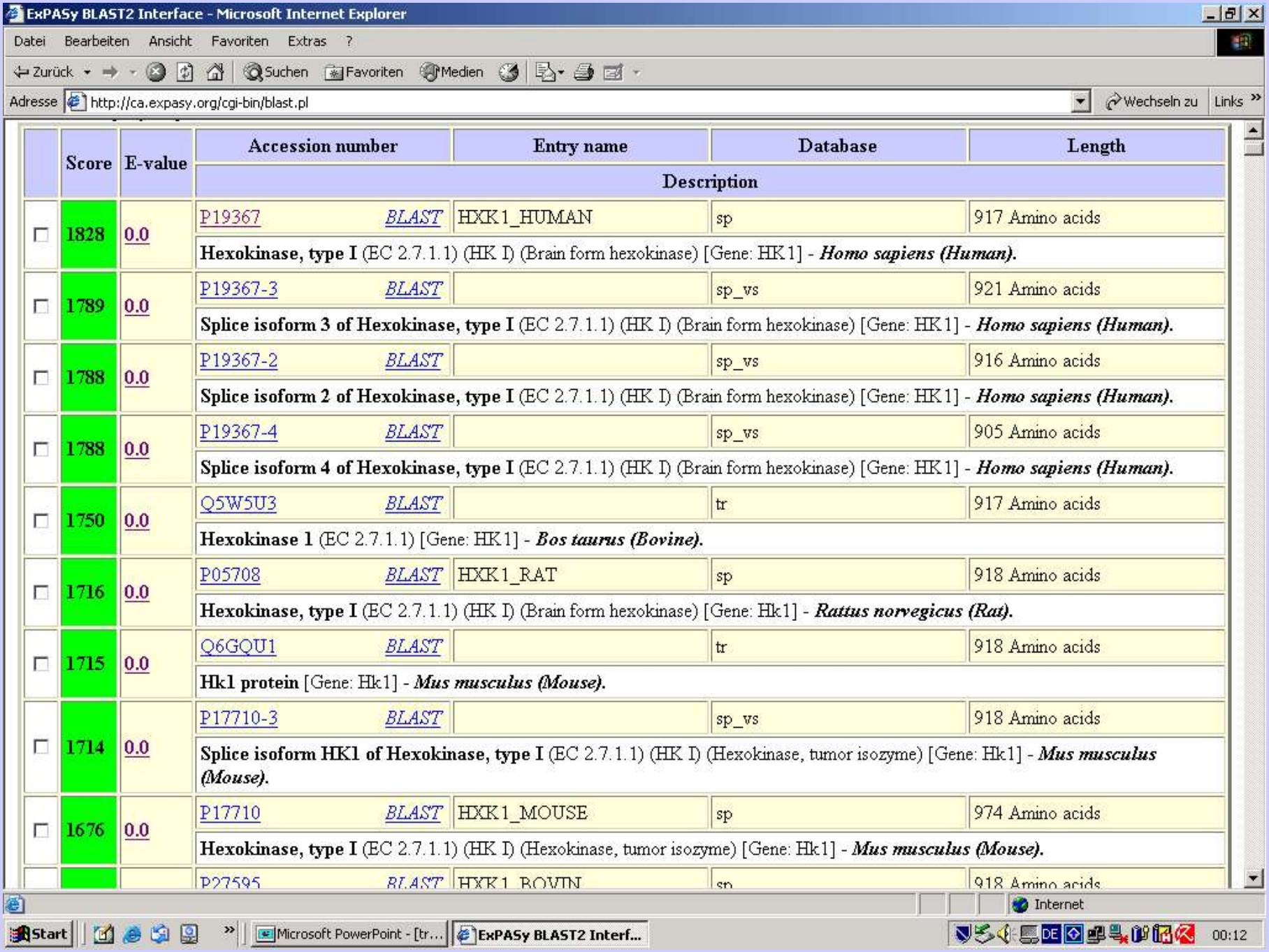
NiceProt View of Swiss-Prot:
P19367

Printer-friendly view | Submit update | Quick BlastP search

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

Note: most headings are clickable, even if they don't appear as links. They link to the user manual or other documents.

Entry information	
Entry name	**HXK1_HUMAN**
Primary accession number	**P19367**
Secondary accession numbers	O43443 O43444 O75574 Q96HC8 Q9NNZ4 Q9NNZ5
Entered in Swiss-Prot in	Release 16, November 1990
Sequence was last modified in	Release 16, November 1990
Annotations were last modified in	Release 45, October 2004
Name and origin of the protein	
Protein name	**Hexokinase, type I**
Synonyms	**EC 2.7.1.1** **HK I** **Brain form hexokinase**
Gene name	**Name: HK1**
From	Homo sapiens (Human) [TaxID: 9606]
Taxonomy	Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi; Mammalia; Eutheria; Primates; Catarrhini; Hominidae; Homo.
References	


ExPASy BLAST2 Interface - Microsoft Internet Explorer
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück Suchen Favoriten Medien
Adresse http://ca.expasy.org/cgi-bin/blast.pl
Wechseln zu Links
Score E-value Accession number Entry name Database Length
Description
1828 0.0 P19367 BLAST HXK1_HUMAN sp 917 Amino acids
Hexokinase, type I (EC 2.7.1.1) (HK I) (Brain form hexokinase) [Gene: HK1] - Homo sapiens (Human).
1789 0.0 P19367-3 BLAST sp_vs 921 Amino acids
Splice isoform 3 of Hexokinase, type I (EC 2.7.1.1) (HK I) (Brain form hexokinase) [Gene: HK1] - Homo sapiens (Human).
1788 0.0 P19367-2 BLAST sp_vs 916 Amino acids
Splice isoform 2 of Hexokinase, type I (EC 2.7.1.1) (HK I) (Brain form hexokinase) [Gene: HK1] - Homo sapiens (Human).
1788 0.0 P19367-4 BLAST sp_vs 905 Amino acids
Splice isoform 4 of Hexokinase, type I (EC 2.7.1.1) (HK I) (Brain form hexokinase) [Gene: HK1] - Homo sapiens (Human).
1750 0.0 Q5W5U3 BLAST tr 917 Amino acids
Hexokinase 1 (EC 2.7.1.1) [Gene: HK1] - Bos taurus (Bovine).
1716 0.0 P05708 BLAST HXK1_RAT sp 918 Amino acids
Hexokinase, type I (EC 2.7.1.1) (HK I) (Brain form hexokinase) [Gene: Hk1] - Rattus norvegicus (Rat).
1715 0.0 Q6GQU1 BLAST tr 918 Amino acids
Hk1 protein [Gene: Hk1] - Mus musculus (Mouse).
1714 0.0 P17710-3 BLAST sp_vs 918 Amino acids
Splice isoform HK1 of Hexokinase, type I (EC 2.7.1.1) (HK I) (Hexokinase, tumor isozyme) [Gene: Hk1] - Mus musculus (Mouse).
1676 0.0 P17710 BLAST HXK1_MOUSE sp 974 Amino acids
Hexokinase, type I (EC 2.7.1.1) (HK I) (Hexokinase, tumor isozyme) [Gene: Hk1] - Mus musculus (Mouse).
P27595 BLAST HXK1_BOVIN sp 918 Amino acids
Internet
Start
Microsoft PowerPoint - [tr...
ExPASy BLAST2 Interf...
00:12

Sum of Squares

A **residue** is the difference between an observed and predicted value of a function. (A predicted value means a value given by some mathematical model.)

Residue = Observed value - Predicted value

The **sum-of-squares error (SSE)** when observed data are approximated by a function is given by

SSE = Sum of squares of residues

= Sum of $(y_{observed} - y_{predicted})^2$

The smaller SSE, the better the approximating function fits the data.

Optimisation

- Newton method
- Steepest Descent
- Levenberg-Marquardt
- Stochastic Methods
- Genetic Algorithms

Newton Method I

Problem: Find $f(x) = 0$

Approach: Develop Taylor series around starting point x_0

$f(x_0 + h) = f(x_0) + hf'(x_0) + 1/2h2f''(x_0)$ …….

Since $f(x_0 + h)$ should be equal 0
$\Rightarrow h = - f(x_0)/f'(x_0)$

The next point is reached with:
$x = x_0 + h = x_0 - f(x_0)/f'(x_0)$

Iteratively: $x^{(k+1)} = x^{(k)} - f(x^{(k)})/f'(x^{(k)})$

Newton Method II

But we want to find $f'(x) = 0$

Therefore:

$f'(x_0 + h) = f'(x_0) + hf''(x_0) + 1/2h2f'''(x_0)$ …….

Since $f'(x_0 + h)$ should be equal 0
$\Rightarrow h = - f'(x_0)/f''(x_0)$

The next point is reached with:
$x = x_0 + h = x_0 - f'(x_0)/f''(x_0)$

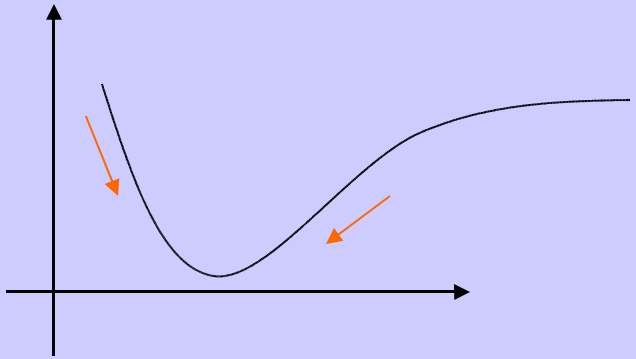
Iteratively: $x^{(k+1)} = x^{(k)} - f'(x^{(k)})/f''(x^{(k)})$

Newton Method III

- Converges quadratically ones it is close to the solution
- Might not converge at all if you are far away
- Sometimes overshoots and overcompensates resulting in oscillations

Steepest Descent

- Simply follows the gradient (first-order derivative) of the function
- Converges linearly and always
- Terribly slow
- You are not able to constrain variables

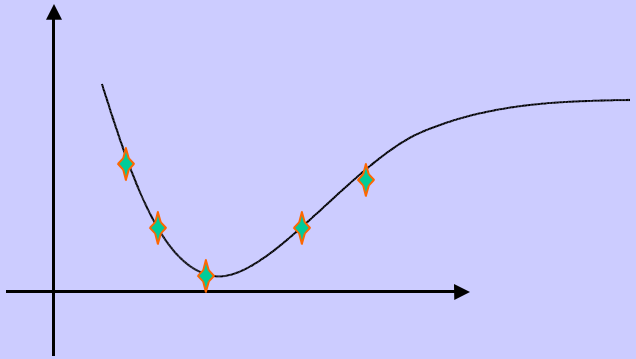


Levenberg-Marquardt

- Hybrid between Newton and Steepest Descent
- If far away from minimum, changes to Steepest Descent
- Widely used and quite powerful

Random Search

- As name indicates tries randomly combinations of parameters and checks the value of the function of interest
- No convergence!
- You need a lot of patience
- Will eventually come up with the global minimum



Direct Search Methods

- Does not rely on computing derivatives
- Local optimizer
- Uses heuristics to „down the function“ based on evaluating the last interation steps

Evolutionary Strategies

- Survival of the „fittest“
- Based on biological inspirations
- GO method

LO vs GO

- Local optimizers are usually faster, but you need to be close to the solution in the beginning
- GO will (if they work) eventually find the global solution, but might take forever
- New developments are hybrid algorithms that use a GO method first and then switch to LO

More Information

Mendes et al., Bioinformatics, Vol. 14, 1998, 869-883

Moles et al., Genome Research, Vol. 13, 2003, 2467-2474

Software: Copasi, www.copasi.org