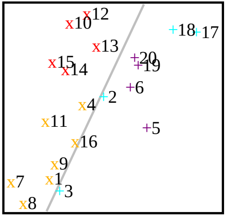
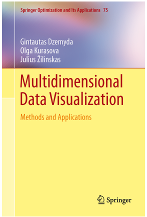
Optimization-Based Visualization of Multidimensional Data

Julius Žilinskas (Юлюс Жилинскас)

Vilnius University, Lithuania

Summer School on Operational Research and Applications
May 16, 2013



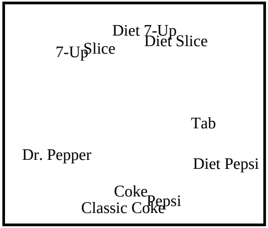


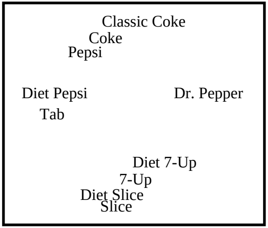
Example of multidimensional data: Pharmacological data

Ligand	$h\alpha_{2A}$	$z\alpha_{2A}$	$h\alpha_{2B}$	$z\alpha_{2B}$	$h\alpha_{2C}$	$z\alpha_{2C}$	$z\alpha_{2Da}$	$z\alpha_{2Db}$
1. Atipamezole	1.6	13	1.5	5.0	4.3	2.1	5.1	6.9
2. Clonidine	10	89	44	250	110	55	120	150
3. Dexmedetomidine	1.3	2.2	4.7	7.6	6.5	12	4.1	3.7
4. Idazoxan	17	85	24	40	17	17	52	94
5. Oxymetazoline	2.1	5.1	1100	1200	130	1300	1100	440
6. UK14,304	32	40	320	1200	190	700	260	280
7. L657.743	0.8	6.9	0.7	1.2	0.09	1.0	1.6	1.3
8. Rauwolscine	1.9	1.0	1.1	1.4	0.2	0.5	2.3	2.3
9. Yohimbine	5.9	5.2	7.5	9.3	4.6	3.4	6.4	4.0
10. Chlorpromazine	990	110	43	1.1	330	83	18	19
11. Clozapine	32	3.3	12	9.3	2.1	3.2	12	24
12. ARC239	2100	1800	9.6	36	66	280	55	44
13. Prazosin	1030	330	66	300	31	100	68	64
14. Spiperone	540	45	12	51	11	63	15	18
15. Spiroxatrine	320	150	2.4	93	3.1	35	11	11
16. WB-4101	5.4	11	60	51	1.9	19	31	16
17. 2-Amino-1-phenylethanol	1300	5400	4200	9400	8100	5100	3700	4000
18. Dopamine	2000	790	6300	4400	1200	3900	1300	1700
19. (−)-Adrenaline	150	140	710	910	130	1080	500	470
20. (−)-Noradrenaline	110	260	680	647	250	580	380	510

Example of multidimensional data: Experimental testing of soft drinks

Soft drinks	1	2	3	4	5	6	7	8	9	10
1. Pepsi	0	127	169	204	309	320	286	317	321	238
2. Coke	127	0	143	235	318	322	256	318	318	231
3. Classic Coke	169	143	0	243	326	327	258	318	318	242
4. Diet Pepsi	204	235	243	0	285	288	259	312	317	194
5. Diet Slice	309	318	326	285	0	155	312	131	170	285
6. Diet 7-Up	320	322	327	288	155	0	306	164	136	281
7. Dr Pepper	286	256	258	259	312	306	0	300	295	256
8. Slice	317	318	318	312	131	164	300	0	132	291
9. 7-Up	321	318	318	317	170	136	295	132	0	297
10. Tab	238	231	242	194	285	281	256	291	297	0





Multidimensional scaling (MDS) – a technique for exploratory analysis of multidimensional data

- Pairwise dissimilarities between n objects are given by a matrix (δ_{ij}), $i, j = 1, \ldots, n$, it is supposed that $\delta_{ij} = \delta_{ji}$.
- The points representing objects in an m-dimensional embedding space $\mathbf{x}_i \in \mathbb{R}^m$, $i = 1, \ldots, n$ should be found whose inter-point distances fit the given dissimilarities.
- The problem is reduced to minimization of a fitness criterion, e.g. so called *Stress* function

$$S(\mathbf{x}) = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \left(d\left(\mathbf{x}_i, \mathbf{x}_j\right) - \delta_{ij} \right)^2 ,$$

where $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$; $d(\mathbf{x}_i, \mathbf{x}_j)$ denotes the distance between the points $\mathbf{x}_i$ and $\mathbf{x}_j$; weights $w_{ij} > 0, i, j = 1, \ldots, n$.

MDS is a difficult global optimization problem

- Although *Stress* function is defined by an analytical formula which seems rather simple, it normally has many local minima.
- The problem is high dimensional: $\mathbf{x} \in \mathbb{R}^N$ and the number of variables is equal to $N = n \times m$.
- *Stress* function is invariant with respect to translation, rotation and mirroring.
- Smoothness of *Stress* function depends on distances $d(\mathbf{x}_i, \mathbf{x}_j)$, however, non-differentiability normally cannot be ignored. Minkowski distances

$$d_r(\mathbf{x}_i, \mathbf{x}_j) = \left(\sum_{k=1}^{m} \left| x_{ik} - x_{jk} \right|^r \right)^{1/r}.$$

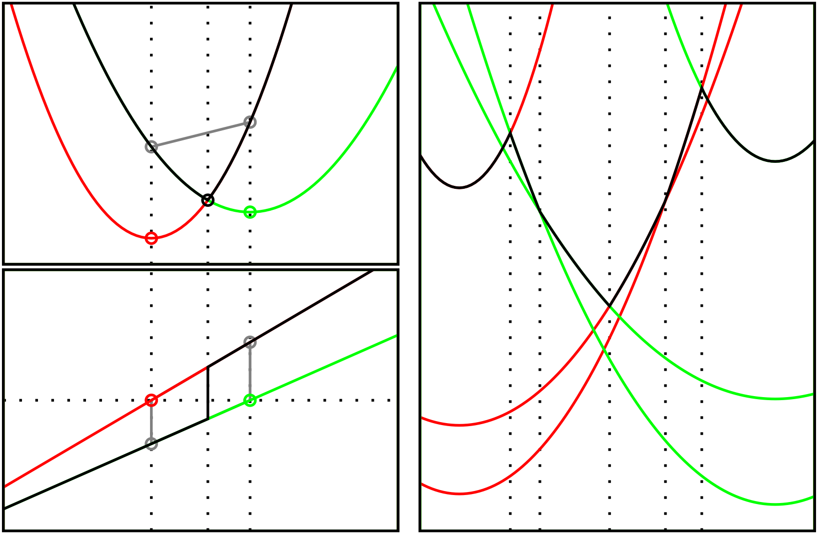
MDS with city-block distances

- If city-block distances $d_1(\mathbf{x}_i, \mathbf{x}_j)$ are used, *Stress* can be redefined as

$$S(\mathbf{x}) = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \left(\sum_{k=1}^{m} \left| x_{ik} - x_{jk} \right| - \delta_{ij} \right)^2 .$$

- In the case of city-block distances and $m \geq 2$ *Stress* can be non-differentiable even at a minimum point. With this respect the case of city-block distances is different from the other cases of Minkowski distances when positiveness of distances $d(\mathbf{x}_i^*, \mathbf{x}_j^*)$, $i, j = 1, \ldots, n$ at a local minimum point $\mathbf{x}^*$ implies differentiability of *Stress*.
- However *Stress* with city-block distances is piecewise quadratic, and such a structure can be exploited for tailoring of ad hoc global optimization algorithms.

Piecewise quadratic function non-differentiable at a minimum point



Decomposition of the original optimization problem into a set of quadratic programming problems

- Let $A(\mathbf{P})$ denote a set such that

  $$A(\mathbf{P}) = \left\{ \mathbf{x} | x_{ik} \leq x_{jk} \text{ for } p_{ki} < p_{kj},\ i,j = 1,\ldots,n,\ k = 1,\ldots,m \right\},$$

  where $\mathbf{P} = (p_{11}, p_{12}, \ldots, p_{mn})$, $\mathbf{p}_k = (p_{k1}, p_{k2}, \ldots, x_{kn})$ are m permutations of $1, \ldots, n$.
- For $\mathbf{x} \in A(\mathbf{P})$,

  $$S(\mathbf{x}) = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \left(\sum_{k=1}^{m} \left(x_{ik} - x_{jk} \right) z_{kij} - \delta_{ij} \right)^2,$$

  where

  $$z_{kij} = \begin{cases} 1, & p_{ki} > p_{kj}, \\ -1, & p_{ki} < p_{kj}. \end{cases}$$

Since function $S(\mathbf{x})$ is quadratic over polyhedron $\mathbf{x} \in A(\mathbf{P})$ the minimization problem

$$\min_{\mathbf{x} \in A(\mathbf{P})} S(\mathbf{x})$$

is a quadratic programming problem. It is equivalent to

$$\min \left[- \sum_{k=1}^{m} \sum_{i=1}^{n} x_{ik} \sum_{j=1}^{n} w_{ij} \delta_{ij} z_{kij} + \right.$$

$$\left. \frac{1}{2} \sum_{k=1}^{m} \sum_{l=1}^{m} \sum_{i=1}^{n} \left(x_{ik} x_{il} \sum_{t=1, t \neq i}^{n} w_{it} z_{kit} z_{lit} - \sum_{j=1, j \neq i}^{n} x_{ik} x_{jl} w_{ij} z_{kij} z_{lij} \right) \right]$$

$$\text{s.t.} \quad \sum_{i=1}^{n} x_{ik} = 0, \; k = 1, \ldots, m,$$

$$x_{\{j | p_{kj} = i+1\}, k} - x_{\{j | p_{kj} = i\}, k} \geq 0, \; k = 1, \ldots, m, \; i = 1, \ldots, n-1.$$

Quadratic programming problem

- The definition of quadratic programming problem in matrix form:

$$\min \left(-\mathbf{d}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{D}\mathbf{x} \right)$$

$$\begin{aligned} \text{s.t. } \mathbf{A}^0\mathbf{x} &= 0, \\ \mathbf{A}^k\mathbf{x} &\geq 0, \; k = 1, \ldots, m, \end{aligned}$$

- where
  - $\mathbf{d}$ is a vector of dimensionality $(nm \times 1)$,
  - $\mathbf{D}$ is an $(nm \times nm)$ matrix,
  - $\mathbf{A}^0$ is an $(m \times nm)$ matrix,
  - $\mathbf{A}^k$, $k = 1, \ldots, m$, are $((n-1) \times nm)$ matrices.

Definition of quadratic programming problem

$$
\begin{aligned}
d_{kn-n+i} &= \sum_{j=1, j\neq i}^{n} w_{ij}\delta_{ij} z_{kij} \,\Bigg|\, \begin{array}{l} k = 1, \ldots, m, \\ i = 1, \ldots, n. \end{array} \\
D_{kn-n+i, ln-n+j} &= \left\{ \begin{array}{ll} \sum_{t=1, t\neq i}^{n} w_{it} z_{kit} z_{lit}, & i = j, \\ -w_{ij} z_{kij} z_{lij}, & i \neq j, \end{array} \right. \Bigg|\, \begin{array}{l} k, l = 1, \ldots, m, \\ i, j = 1, \ldots, n. \end{array} \\
A^0_{kj} &= \left\{ \begin{array}{ll} 1, & j = kn - n + 1, \ldots, kn, \\ 0, & \text{otherwise}, \end{array} \right. \Bigg|\, \begin{array}{l} k = 1, \ldots, m, \\ j = 1, \ldots, mn. \end{array} \\
A^k_{ij} &= \left\{ \begin{array}{ll} 1, & p_{k, j-kn+n} = i + 1, \\ -1, & p_{k, j-kn+n} = i, \\ 0, & \text{otherwise}, \end{array} \right. \Bigg|\, \begin{array}{l} k = 1, \ldots, m, \\ i = 1, \ldots, n-1, \\ j = 1, \ldots, mn. \end{array}
\end{aligned}
$$

Two level MDS method with city-block distances

- Taking into account the structure of the minimization problem a two level minimization method can be applied: to solve a combinatorial optimization problem at the upper level, and to solve a quadratic programming problem at the lower level:

$$\min_{\mathbf{P}} S(\mathbf{P}), \text{ s.t. } S(\mathbf{P}) = \min_{\mathbf{x} \in A(\mathbf{P})} S(\mathbf{x}) \sim$$

$$\sim \min \left(-\mathbf{c_P}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{Q_P} \mathbf{x} \right) \text{ s.t. } \mathbf{E}\mathbf{x} = \mathbf{0}, \ \mathbf{A_P}\mathbf{x} \geq \mathbf{0},$$

$$\mathbf{c_P} \in \mathbb{R}^{nm}, \ \mathbf{Q_P} \in \mathbb{R}^{nm \times nm}, \ \mathbf{E} \in \mathbb{R}^{m \times nm}, \ \mathbf{A_P} \in \mathbb{R}^{(n-1) \times nm}.$$

- For the lower level problem a quadratic programming method can be applied.

Upper level combinatorial problem

- The upper level objective function is defined over the set of m-tuple of permutations of $1, ..., n$.
- The number of feasible solutions is $(n!)^m$, avoiding mirrored solutions it can be reduced to approximately $(n!)^m/(2^m m!)$.
- Solution of combinatorial problem:
  - explicit enumeration of all feasible solutions;
  - branch and bound;
  - pure random search;
  - multistart;
  - evolutionary algorithm.

Local search based on quadratic programming

- A minimum point of a quadratic programming problem is not necessary a local minimizer of the initial problem of minimization of *Stress*, if it is on the boundary of polyhedron.
- Therefore local search may be continued:
  - Go to the neighbor polyhedron on the opposite side of the active inequality constrains.
  - If $i, \ldots, j$ inequality constraints $\mathbf{A}_{\mathrm{P}}\mathbf{x} \geq \mathbf{0}$ are active, $i \leq p_{kt} \leq j+1$ should be updated to $i+j+1-p_{kt}$.
  - Perform quadratic programming.
  - Repeat while better values are found and some inequality constrains are active.

Parallel explicit enumeration

- Each processor runs the same algorithm generating feasible solutions which should be enumerated explicitly, but only each p-th is explicitly enumerated on a processor where p is the number of processors.
- The first processor explicitly enumerates the first, $(p+1)$ and so on generated solutions. The second processor enumerates 2-nd, $(p+2)$, ... The p-th processor enumerates p-th, $2p$-th, ...
- It is assumed that generation of the solutions to be explicitly enumerated requires much less computational time than the explicit enumeration which requires solution of the lower level quadratic programming problem.
- The results of different processors are collected when the generation of solutions and explicit enumeration are finished.
- The standardized message-passing communication protocol MPI is used for communication between parallel processors.

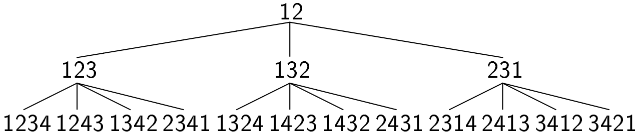
Branch and bound for MDS with city-block distances

- A subset of feasible solutions is represented by a partial solution defined by m-tuple $\overline{\mathbf{P}}$ of permutations of $1, \dots, \overline{n}$ where only $\overline{n}$ of n objects are considered.
- Depth first selection strategy is used to avoid storing of candidate nodes.
- The lower bound is a partial *Stress* evaluated at the solution of the lower level quadratic programming problem for $\overline{n}$ objects over a polyhedron $A(\overline{\mathbf{P}})$:

$$\min_{\overline{\mathbf{x}} \in A(\overline{\mathbf{P}})} \sum_{i=1}^{\overline{n}} \sum_{j=1}^{\overline{n}} w_{ij} \left(\sum_{k=1}^{m} \left| x_{ik} - x_{jk} \right| - \delta_{ij} \right)^2,$$

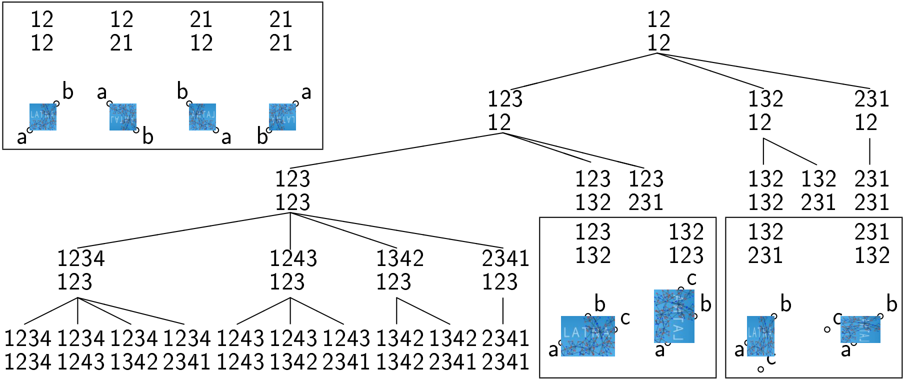
where $\overline{\mathbf{x}} = (\mathbf{x}_1, \dots, \mathbf{x}_{\overline{n}})$.

A search tree for $m = 1$



- Every numeral represents a value of p_{1i}, $i = 1, \ldots, \overline{n}$.
- To refuse mirrored solutions the search tree starts with a root node representing partial solution "12". Although the sequence numbers of the first two objects may be changed (13 and 23) after assignment of the third object, their sequence is not changed ($1 < 2$, $1 < 3$ and $2 < 3$).
- The number of feasible solutions is $n!/2$.
- The number of nodes $\sum_{i=2}^{n} \frac{i!}{2}$.

A search tree for $m = 2$



- The number of feasible solutions is $n!^3/48 + n!^2/8 + n!/6$.

Parallel branch and bound algorithm

- Each process runs the same algorithm generating partial solutions of up to some level l: $\overline{n} \leq l$.
- Each p-th partial solution at the level $\overline{n} = l$ is evaluated and branched if needed, where p is the number of processes.
- The first process evaluates and branches the first, $(p+1)$ and so on generated partial solutions. The second process evaluates 2-nd, $(p+2)$, ... The p-th processor evaluates p-th, $2p$-th, ...
- The results of different processes are collected at the end of computation.
- The standardized message-passing communication protocol MPI is used for communication between parallel processes.

Evolutionary algorithm (n_p, t_c, N_{init}, p_{mut})

Generate N_{init} random feasible solutions.
Perform local search from n_p best solutions to form initial population.
while t_c time has not passed
    With probability p_{mut} mutate randomly chosen individual.
    Uniformly randomly generate two indexes of parents i and j.
    Uniformly randomly generate two integers k and l, $k, l = 1, \dots, n$.
    Compose permutations of descendant:
        elements $1, \dots, k-1, l+1, \dots, n$ are elements of $\mathbf{p}_i$,
        elements $k, \dots, l$ are the missing numbers ordered in the
            same way as they are ordered in $\mathbf{p}_j$.
    Perform local search based on quadratic programming.
    If the offspring is more fitted than the worst individual,
        then the offspring replaces the latter.

Parallel version of evolutionary algorithm

- Multiple populations.
- Each processor runs the same evolutionary algorithm with different sequences of random numbers.
- The results of different processors are collected when search is finished after predefined time.
- Communications between processors are kept to minimum.

Experimental investigation

- The accuracy of fit evaluated via minimum of $S(\mathbf{x})$ depends on n and $\delta_{ij}, i, j = 1, \ldots, n$. To reduce this undesirable impact, a relative error is used in the presentation of the results:

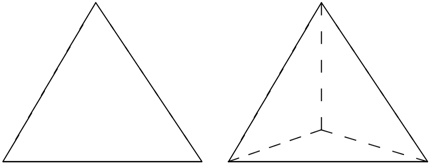
$$f(\mathbf{x}) = \sqrt{S(\mathbf{x}) \Big/ \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \delta_{ij}^2}.$$

- Performance is measured using
  - Time and number of quadratic programming problems solved;
  - Mean ($\overline{f^*}$) and standard deviation (s.d. f^*) of global minimum estimates;
  - Best estimate of global minimum (f^*) and percentage of runs ($perc$) when the estimate differs from f^* by less than 10^{-4}.
- Various data sets.

Data set: vertices of the standard simplex

- The distances between any two vertices are equal. $n = \dim + 1$.

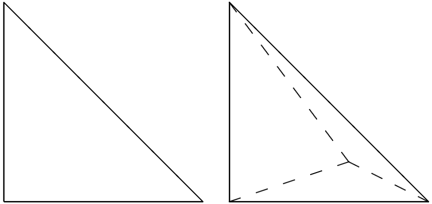
$$\Delta = \begin{pmatrix} 0 & 1 & 1 & 1 & \cdots & 1 & 1 \\ 1 & 0 & 1 & 1 & & 1 & 1 \\ 1 & 1 & 0 & 1 & & 1 & 1 \\ 1 & 1 & 1 & 0 & & 1 & 1 \\ \vdots & & & & \ddots & & \vdots \\ 1 & 1 & 1 & 1 & \cdots & 0 & 1 \\ 1 & 1 & 1 & 1 & \cdots & 1 & 0 \end{pmatrix}.$$



Data set: vertices of the unit simplex

$$v_{ij} = \left\{ \begin{array}{ll} 1, & i = j+1, \\ 0, & \text{otherwise}, \end{array} \right| i = 1, \ldots, n, j = 1, \ldots, dim. \; n = \dim + 1.$$

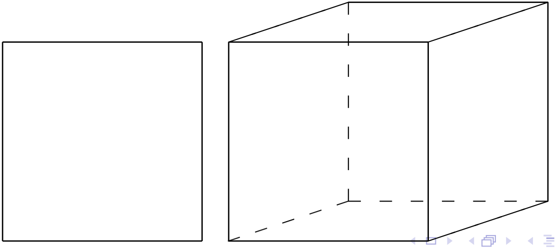
$$\Delta^1 = \begin{pmatrix} 0 & 1 & 1 & \cdots & 1 \\ 1 & 0 & 2 & & 2 \\ 1 & 2 & 0 & & 2 \\ \vdots & & & \ddots & \vdots \\ 1 & 2 & 2 & \cdots & 0 \end{pmatrix}, \; \Delta^2 = \begin{pmatrix} 0 & 1 & 1 & \cdots & 1 \\ 1 & 0 & \sqrt{2} & & \sqrt{2} \\ 1 & \sqrt{2} & 0 & & \sqrt{2} \\ \vdots & & & \ddots & \vdots \\ 1 & \sqrt{2} & \sqrt{2} & \cdots & 0 \end{pmatrix}.$$



Data set: vertices of multidimensional cube

- The coordinates of i-th vertex of a dim-dimensional cube are equal either to 0 or to 1, and they are defined by binary code of $i = 0, \ldots, n-1$. $n = 2^{\mathrm{dim}}$.

$$\Delta^1 = \begin{pmatrix} 0 & 1 & 1 & 2 & \cdots & n-1 & n \\ 1 & 0 & 2 & 1 & & n & n-1 \\ 1 & 2 & 0 & 1 & & n-2 & n-1 \\ 2 & 1 & 1 & 0 & & n-1 & n-2 \\ \vdots & & & & \ddots & & \vdots \\ n-1 & n & n-2 & n-1 & & 0 & 1 \\ n & n-1 & n-1 & n-2 & \cdots & 1 & 0 \end{pmatrix}.$$



Data set: error-perturbed distance data

- The data generated using uniformly distributed random points in m-dimensional space whose pairwise dissimilarities are computed by
$$\delta_{ij} = \sum_{k=1}^{m} \left| x_{ik}^{(e)} - x_{jk}^{(e)} \right|,$$
where $x_{ik}^{(e)} = x_{ik} + N(0, e(x_{ik}))$, and $N(0, e)$ denotes the normally distributed random variable with mean zero and standard deviation e.
- Eight problems defined by all combinations of the parameters $(n = 10, 20;\ m = 2, 3,\ e(x) = 0.15x, 0.3x)$ have been generated and they are referred as 'ghm'.

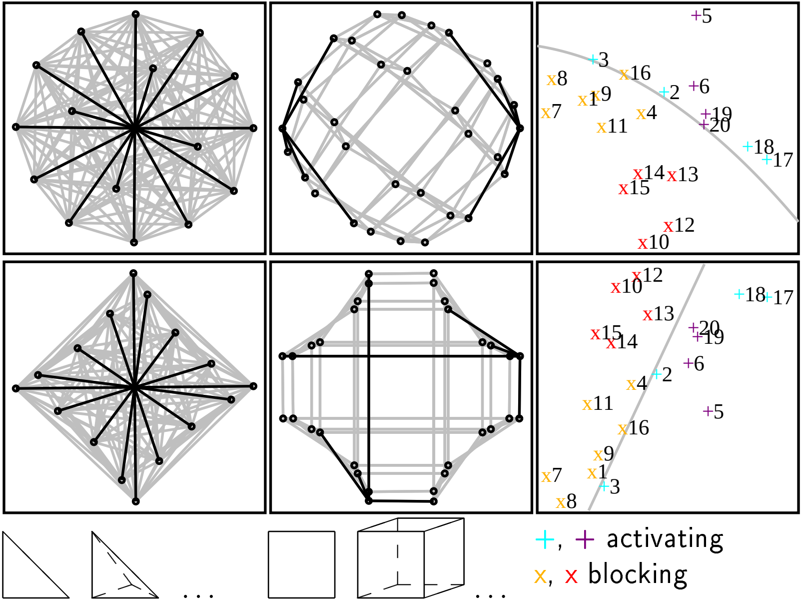
Empirical data sets

- A frequently used test problem for MDS algorithms is based on experimental testing of $n = 10$ soft drinks, where dissimilarities were measured by means of psychological experiment. This problem is referred as 'cola'.
- Another frequently used test problem is confusion data for ($n = 36$) Morse codes. This problem is referred as 'morsecodes'.

Pharmacological binding affinity data

- Binding affinity data is represented through a matrix, one dimension formed by different ligands tested in a series of experiments while the other represents different proteins.
- Receptor proteins can be from different types or subtypes, or from different species, or engineered mutants of these.
- Ligands can be natural neurotransmitters or pharmacological drugs, an agonist activates, an antagonist blocks the receptor.
- Dissimilarities are computed as distances between vectors of the $\log_{10}$-transformed binding affinities.
  - Three human and five zebrafish α_2-adrenoceptor proteins, and 20 ligands (Ruuskanen et al., 2005);
  - human, rat, guinea pig and pig proteins (Uhlen et al., 1998);
  - wild type and mutant proteins (Hwa et al., 1995).

MDS with Euclidean and city-block distances



The best known estimates of global minimum

datasets		n	$\min f^*,\ m=2$	$\min f^*,\ m=3$
standard		8	**0.2825**	**0.1250**
simplices		12	**0.3300**	0.2013
		16	0.3525	0.2321
		20	0.3657	0.2508
unit		8	**0.2569**	**0.0992**
simplices		12	**0.3167**	0.1874
		16	0.3439	0.2243
		20	0.3595	0.2459
cubes		8	**0.2245**	**0.00**
		16	0.2965	0.1590
		32	0.3313	0.2078
	$e\%$			
ghm	15	10	0.1293	0.0906
	30	10	0.2711	0.1298
	15	20	0.1868	0.1610
	30	20	0.2966	0.2284
cola		10	**0.1647**	0.0659
morsecodes		36	0.2944	0.1962

Reduced search space of problems exposing symmetries

- If exchange of some objects does not change dissimilarity data, exchange of points representing these objects does not change the value of *Stress* function.
- In continuous optimization: constrain sequence of first coordinate values of image points of symmetric objects.
- In combinatorial optimization: allow only some of permutations of first coordinate $\mathbf{p}_1$.
- For geometric data sets
  - standard simplex: $x_{11} \leq x_{21} \leq \ldots \leq x_{n1}$; $\mathbf{p}_1 = (1, 2, \ldots, n)$.
  - unit simplex: $x_{21} \leq x_{31} \leq \ldots \leq x_{n1}$; $\mathbf{p}_1 = (l, 1, 2, \ldots, l-1, l+1, \ldots, n)$.
  - hyper-cube: $x_{11} \leq x_{i1}, i = 2, \ldots n$; $p_{11} = 1$.

Explicit enumeration: standard simplices

	$m=1$		$m=2$		$m=3$	
n	t, s (NQP)	f^*	t, s (NQP)	f^*	t, s (NQP)	f^*
3	0.00 (3)	0.3333	0.00 (6)	0.00	0.00 (10)	0.00
4	0.00 (12)	0.4082	0.00 (78)	0.00	0.01 (364)	0.00
5	0.00 (60)	0.4472	0.03 (1830)	0.1907	1.79 (37820)	0.00
6	0.00 (360)	0.4714	1.71 (64980)	0.2309	589.72 (7840920)	0.00
7	0.03 (2520)	0.4879	118.59 (3176460)	0.2621		
8	0.21 (20160)	0.5000	10229.00 (203222880)	0.2825		
9	2.24 (181440)	0.5092				
10	26.63 (1814400)	0.5164				
11	351.09 (19958400)	0.5222				
12	4702.00 (239500800)	0.5270				
3	0.00 (1)	0.3333	0.00 (3)	0.00	0.00 (6)	0.00
4	0.00 (1)	0.4082	0.00 (12)	0.00	0.00 (78)	0.00
5	0.00 (1)	0.4472	0.00 (60)	0.1907	0.09 (1830)	0.00
6	0.00 (1)	0.4714	0.01 (360)	0.2309	5.01 (64980)	0.00
7	0.00 (1)	0.4879	0.10 (2520)	0.2621	379.88 (3176460)	0.094
8	0.00 (1)	0.5000	1.01 (20160)	0.2825	31681.00 (203222880)	0.125
9	0.00 (1)	0.5092	11.89 (181440)	0.2991		
10	0.00 (1)	0.5164	153.88 (1814400)	0.3115		
11	0.00 (1)	0.5222	2121.56 (19958400)	0.3217		
12	0.00 (1)	0.5270	31170.00 (239500800)	0.3300		

Explicit enumeration: unit simplices

	$m = 1$		$m = 2$		$m = 3$	
n	t, s (NQP)	f^*	t, s (NQP)	f^*	t, s (NQP)	f^*
3	0.00 (3)	0.00	0.00 (6)	0.00	0.00 (10)	0.0
4	0.00 (12)	0.3651	0.00 (78)	0.00	0.01 (364)	0.0
5	0.00 (60)	0.4140	0.04 (1830)	0.00	2.02 (37820)	0.0
6	0.01 (360)	0.4554	2.05 (64980)	0.1869	661.11 (7840920)	0.0
7	0.02 (2520)	0.4745	137.12 (3176460)	0.2247		
8	0.23 (20160)	0.4917	11662.00 (203222880)	0.2569		
9	2.51 (181440)	0.5018				
10	29.78 (1814400)	0.5113				
11	378.45 (19958400)	0.5176				
12	5265.00 (239500800)	0.5236				
3	0.00 (2)	0.00	0.00 (4)	0.00	0.00 (7)	0.0
4	0.00 (2)	0.3651	0.00 (18)	0.00	0.01 (99)	0.0
5	0.00 (3)	0.4140	0.01 (108)	0.00	0.14 (2574)	0.0
6	0.00 (3)	0.4554	0.02 (720)	0.1869	8.49 (101160)	0.0
7	0.00 (4)	0.4745	0.25 (5760)	0.2247	695.19 (5446080)	0.0
8	0.00 (4)	0.4917	2.90 (50400)	0.2569	66686.00 (381049200)	0.09
9	0.00 (5)	0.5018	37.16 (504000)	0.2759		
10	0.00 (5)	0.5113	560.84 (5443200)	0.2936		
11	0.00 (6)	0.5176	7813.00 (65318400)	0.3058		
12	0.00 (6)	0.5236	122360.00 (838252800)	0.3167		

Explicit enumeration: hyper-cubes

	$m = 1$		$m = 2$		$m = 3$	
n	t, s (NQP)	f^*	t, s (NQP)	f^*	t, s (NQP)	f^*
4	0.00 (12)	0.4082	0.00 (78)	0.00	0.02 (364)	0.00
8	0.24 (20160)	0.4787	12572.00 (203222880)	0.2245		
4	0.00 (6)	0.4082	0.00 (57)	0.00	0.01 (308)	0.00
8	0.06 (5040)	0.4787	5483.00 (88908120)	0.2245		

Explicit enumeration on SUN Fire E15k parallel computer: simplices, $n = 7$, $m = 2$

	standard simplex			unit simplex		
p	t, s	s_p	e_p	t, s	s_p	e_p
1	1037	1.00	1.00	1299	1.00	1.00
2	518	2.00	1.00	650	2.00	1.00
3	349	2.97	0.99	438	2.97	0.99
4	261	3.97	0.99	327	3.97	0.99
5	210	4.95	0.99	262	4.95	0.99
6	175	5.91	0.98	219	5.93	0.99
7	151	6.88	0.98	188	6.89	0.98
8	134	7.73	0.97	168	7.75	0.97
9	118	8.80	0.98	147	8.85	0.98
10	107	9.71	0.97	134	9.66	0.97
11	97	10.72	0.97	120	10.78	0.98
12	90	11.58	0.96	111	11.69	0.97
13	82	12.62	0.97	102	12.68	0.98
14	77	13.54	0.97	95	13.62	0.97
15	72	14.39	0.96	89	14.55	0.97
16	67	15.44	0.97	84	15.54	0.97
17	64	16.12	0.95	79	16.47	0.97
18	60	17.23	0.96	76	17.18	0.95
19	58	17.90	0.94	72	18.03	0.95
20	55	18.95	0.95	68	19.17	0.96
21	52	19.96	0.95	65	19.96	0.95
22	49	20.95	0.95	62	20.85	0.95
23	48	21.81	0.95	59	22.06	0.96
24	46	22.50	0.94	57	22.60	0.94

Worst case results of branch and bound: simplices, $m = 1$

	n	f^*	Enumeration: t, s (NQPP)	Branch and bound: t, s (NQPP)
standard	3	0.3333	0.00 (3)	0.00 (3)
simplices	4	0.4082	0.00 (12)	0.00 (14)
	5	0.4472	0.00 (60)	0.00 (73)
	6	0.4714	0.01 (360)	0.01 (432)
	7	0.4879	0.02 (2520)	0.05 (2951)
	8	0.5000	0.21 (20160)	0.24 (23110)
	9	0.5092	2.22 (181440)	2.47 (204549)
	10	0.5164	27.39 (1814400)	28.33 (2018948)
	11	0.5222	334.30 (19958400)	361.60 (21977347)
	12	0.5270	4687.0 (239500800)	4970.0 (261478146)
	13	0.5311	68762 (3113510400)	73714 (3374988545)
unit	3	0.00	0.00 (3)	0.00 (3)
simplices	4	0.3651	0.00 (12)	0.00 (14)
	5	0.4140	0.00 (60)	0.00 (73)
	6	0.4554	0.00 (360)	0.01 (432)
	7	0.4745	0.04 (2520)	0.03 (2951)
	8	0.4917	0.24 (20160)	0.32 (23110)
	9	0.5018	2.48 (181440)	2.77 (204549)
	10	0.5113	33.16 (1814400)	31.67 (2018948)
	11	0.5176	372.59 (19958400)	404.64 (21977347)
	12	0.5236	5208.0 (239500800)	5545.0 (261478146)
	13	0.5279	78579 (3113510400)	86436 (3374988545)

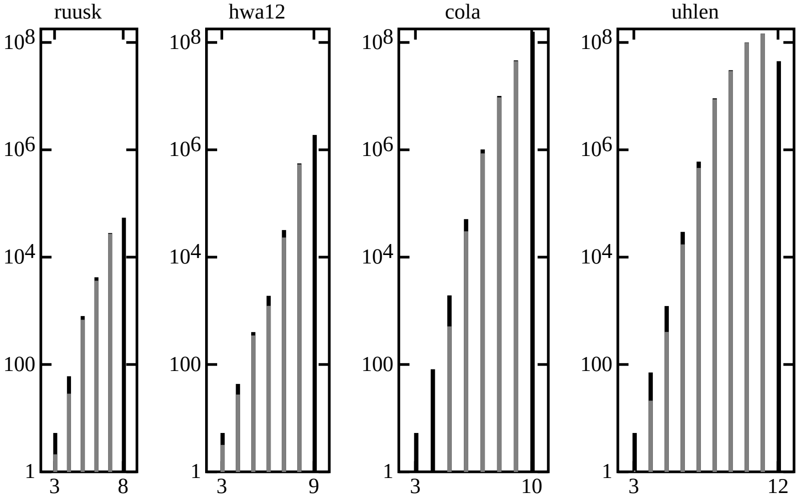
Pessimistic results of branch and bound: simplices, $m > 1$

	m	n	f^*	Enumeration: t, s (NQPP)	Branch and bound: t, s (NQPP)
standard	2	4	0.00	0.01 (78)	0.00 (63)
simplices	2	5	0.1907	0.05 (1830)	0.03 (1322)
	2	6	0.2309	1.73 (64980)	0.85 (27255)
	2	7	0.2621	113.77 (3176460)	59.61 (1655631)
	2	8	0.2825	10183 (203222880)	5107.0 (102073658)
	2	9	0.2991		502844 (3574743410)
	3	4	0.00	0.02 (364)	0.01 (133)
	3	5	0.00	1.79 (37820)	1.12 (23017)
	3	6	0.00	580.87 (7840920)	25.49 (335771)
	3	7	0.0945	301860 (2670344040)	11111 (92710201)
unit	2	4	0.00	0.00 (78)	0.00 (73)
simplices	2	5	0.00	0.05 (1830)	0.03 (662)
	2	6	0.1869	2.02 (64980)	0.51 (16076)
	2	7	0.2247	133.28 (3176460)	17.65 (422940)
	2	8	0.2569	11631 (203222880)	1675.0 (29943080)
	2	9	0.2759		134281 (1905072549)
	3	4	0.00	0.02 (364)	0.02 (313)
	3	5	0.00	2.02 (37820)	0.49 (9837)
	3	6	0.00	653.91 (7840920)	46.67 (578691)
	3	7	0.00	334788 (2670344040)	2652.0 (20674563)

Realistic results of branch and bound: cubes and empirical datasets

	m	n	f^*	Enumeration: t, s (NQPP)	Branch and bound: t, s (NQPP)
cubes	1	4	0.4082	0.00 (12)	0.00 (14)
	1	8	0.4787	0.23 (20160)	0.12 (11260)
	2	4	0.00	0.00 (78)	0.00 (73)
	2	8	0.2245	12518 (203222880)	124.68 (2157090)
	3	4	0.00	0.02 (364)	0.02 (353)
	3	8	0.00		6189 (35216122)
ruusk	1	8	0.2975	0.25 (20160)	0.02 (665)
	2	8	0.1096	12183 (203222880)	3.85 (82617)
	3	8	0.0188		838.68 (6381457)
hwa12	1	9	0.0107	3.06 (181440)	0.02 (2217)
	2	9	0.00		203.25 (2344833)
cola	1	10	0.3688	27.47 (1814400)	1.46 (65642)
	2	10	0.1647		14136 (163235214)
uhlen	1	12	0.2112	6413.0 (239500800)	0.62 (36559)
	2	12	0.0825		35951 (312924750)
hwa21	1	12	0.1790	6648.0 (239500800)	1.49 (71748)

Histograms of levels of solutions



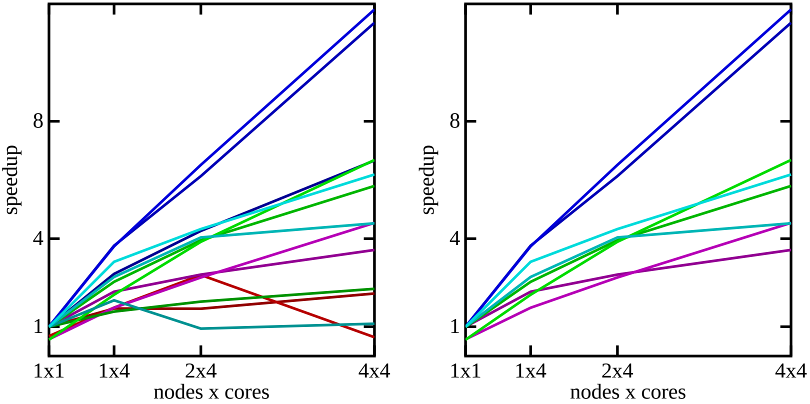
Results of parallel algorithm

l	1 × 1 t, s	1 × 1 NQP	1 × 4 t, s	1 × 4 NQP	2 × 4 t, s	2 × 4 NQP	4 × 4 t, s	4 × 4 NQP
0	2.28	82617		ruusk1, $n = 8$, $m = 2$				
4	2.28	83630	1.41	180637	1.41	288547	1.07	475328
5	3.36	142800	1.38	191839	0.83	240849	3.56	302056
0	479.41	6381457		ruusk1, $n = 8$, $m = 3$				
4	479.69	6403810	218.62	10085145	172.68	14201943	132.61	21415601
5	840.87	13887913	291.15	16040461	179.02	18143183	105.64	20847563
0	120.95	2344833		hwa12, $n = 9$, $m = 2$				
4	120.84	2346237	79.52	5138135	64.96	6986355	52.61	10742968
5	121.89	2407591	47.64	3135785	30.55	3699097	20.79	5014227
6	216.78	5488888	57.66	5565997	30.95	5842364	18.04	6199196
7							612.00	208871963
0	9032	204022569		cola, $n = 10$, $m = 2$				
4	9011	204022487	3212	229324265	2108	270022713	1349	326420931
5	8991	204037437	2391	212954615	1466	226026528	792	244647590
6	8999	206189960	2405	212379122	1380	224377631	761	241643940
7	15189	396725753	4607	410841540	2700	433396504	1388	438645561
0	20515	312924750		uhlen1, $n = 12$, $m = 2$				
4	20494	312925348	10754	556642796	21847	1278648079	18532	2149661364
5	20428	312960596	7579	386113225	5054	508285836	4516	751368922
6	20522	315503838	6360	363849948	4721	461364157	3302	570468417
7	27674	506947703	17241	947746934	47190	2370684051	28521	3044562204

Speedup

$$s_p = \frac{t_1}{t_p},$$

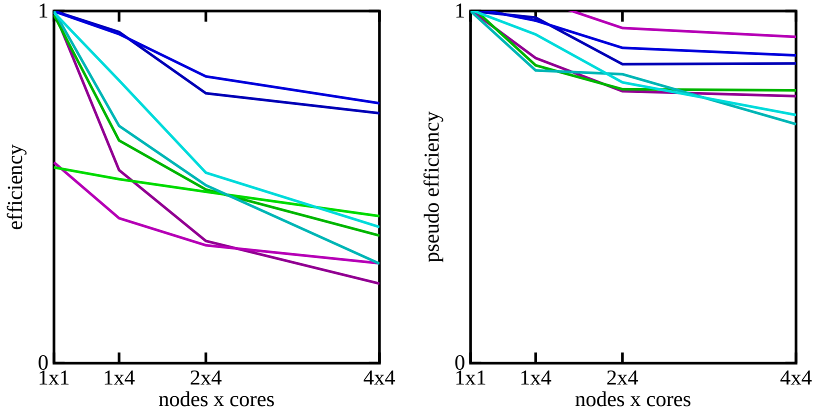
t_p is the time used by the algorithm implemented on p cores, t_1 is the shortest time on one process.



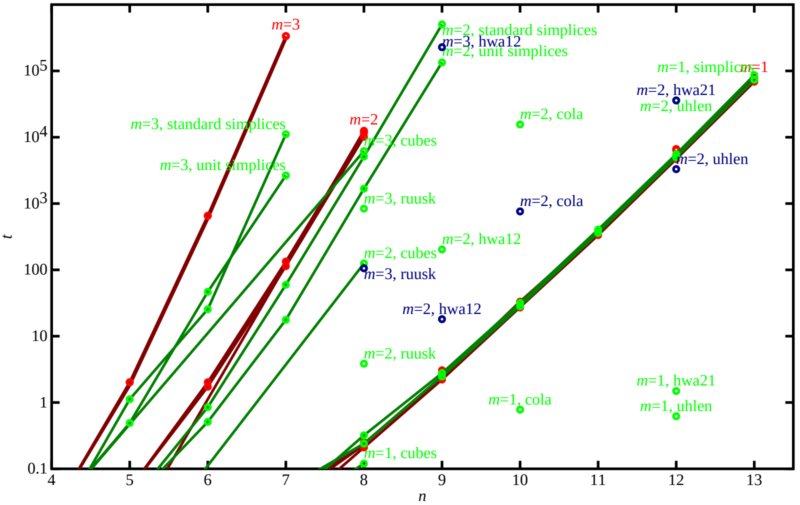
Efficiency of parallelization

$$e_p = \frac{s_p}{p} = \frac{t_1}{p \times t_p}, \quad pe_p = \frac{t_1 / T_1}{p \times t_p / T_p},$$

T_p is the measure of amount of work done (NQP).



Results of explicit enumeration and branch and bound



Random search, multistart and evolutionary algorithms: $m = 2,\ t_c = 10s$

datasets			random search		multistart		evolutionary	
		n	$\overline{f^*}$	s.d. f^*	$\overline{f^*}$	s.d. f^*	$\overline{f^*}$	s.d. f^*
standard		8	0.2825	0.0000	0.2825	0.0000	0.2825	0.0000
simplices		12	0.3326	0.0008	0.3310	0.0004	0.3301	0.0002
		16	0.3575	0.0009	0.3550	0.0005	0.3530	0.0004
		20	0.3720	0.0011	0.3686	0.0004	0.3663	0.0003
unit		8	0.2569	0.0000	0.2569	0.0000	0.2569	0.0000
simplices		12	0.3218	0.0015	0.3168	0.0002	0.3167	0.0000
		16	0.3527	0.0016	0.3463	0.0006	0.3440	0.0002
		20	0.3701	0.0019	0.3627	0.0005	0.3597	0.0002
cubes		8	0.2304	0.0091	0.2245	0.0000	0.2245	0.0000
		16	0.3857	0.0095	0.3012	0.0021	0.2966	0.0002
		32	0.4753	0.0056	0.3508	0.0060	0.3346	0.0021
	e%							
ghm	15	10	0.1695	0.0083	0.1293	0.0000	0.1293	0.0000
	30	10	0.3084	0.0084	0.2711	0.0000	0.2711	0.0000
	15	20	0.3708	0.0166	0.1872	0.0005	0.1868	0.0000
	30	20	0.4282	0.0085	0.3034	0.0025	0.2967	0.0005
cola		10	0.1983	0.0074	0.1667	0.0012	0.1648	0.0003
morsecodes		36	0.4073	0.0040	0.3329	0.0063	0.3125	0.0048

Evolutionary algorithm: $m = 2$, $t_c = 10s$, $n_p = 60$

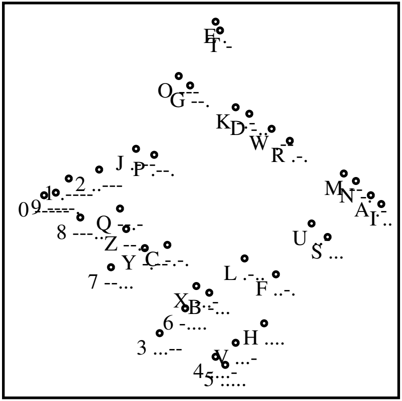
datasets			$N_{init} = 60$ $p_{mut} = 0.00$		$N_{init} = 100$ $p_{mut} = 0.00$		$N_{init} = 100$ $p_{mut} = 0.01$	
		n	$\overline{f^*}$	s.d. f^*	$\overline{f^*}$	s.d. f^*	$\overline{f^*}$	s.d. f^*
standard		8	0.2825	0.0000	0.2825	0.0000	0.2825	0.0000
simplices		12	0.3301	0.0002	0.3300	0.0002	0.3300	0.0001
		16	0.3530	0.0004	0.3529	0.0004	0.3527	0.0003
		20	0.3663	0.0003	0.3661	0.0002	0.3661	0.0003
unit		8	0.2569	0.0000	0.2569	0.0000	0.2569	0.0000
simplices		12	0.3167	0.0000	0.3167	0.0000	0.3167	0.0000
		16	0.3440	0.0002	0.3440	0.0001	0.3440	0.0001
		20	0.3597	0.0002	0.3596	0.0002	0.3596	0.0002
cubes		8	0.2245	0.0000	0.2245	0.0000	0.2245	0.0000
		16	0.2966	0.0002	0.2966	0.0002	0.2966	0.0001
		32	0.3346	0.0021	0.3354	0.0029	0.3355	0.0028
	e%							
ghm	15	10	0.1293	0.0000	0.1293	0.0000	0.1293	0.0000
	30	10	0.2711	0.0000	0.2711	0.0000	0.2711	0.0000
	15	20	0.1868	0.0000	0.1868	0.0000	0.1868	0.0000
	30	20	0.2967	0.0005	0.2968	0.0008	0.2970	0.0012
cola		10	0.1648	0.0003	0.1651	0.0006	0.1651	0.0006
morsecodes		36	0.3125	0.0048	0.3061	0.0027	0.3057	0.0028

Evolutionary algorithm and distance smoothing

datasets				evolutionary algorithm		distance smoothing	
	$e\%$	n	m	$\overline{f^*}$	s.d. f^*	$\overline{f^*}$	s.d. f^*
ghm	15	10	2	**0.1293**	0.0000	0.1457	0.0150
	30	10	2	**0.2711**	0.0000	0.2878	0.0113
	15	20	2	**0.1868**	0.0000	0.2071	0.0130
	30	20	2	**0.2970**	0.0012	0.3093	0.0076
cola		10	2	**0.1651**	0.0006	0.1823	0.0136
morsecodes		36	2	**0.3057**	0.0028	0.3106	0.0966
ghm	15	10	3	**0.0906**	0.0000	0.1116	0.0146
	30	10	3	**0.1298**	0.0000	0.1486	0.0086
	15	20	3	**0.1629**	0.0016	0.1761	0.0065
	30	20	3	**0.2394**	0.0031	0.2454	0.0063
cola		10	3	**0.0673**	0.0013	0.0914	0.0087
morsecodes		36	3	0.2231	0.0055	**0.2045**	0.0062

Comparison of evolutionary algorithm with simulated annealing: morsecodes, $m = 2$

min $S^*/2$	max $S^*/2$	time (s)
evolutionary algorithm		
153.5395	154.5550	1000
153.1380	154.0815	2000
153.0355	153.9175	10000
simulated annealing		
153.2583	155.2006	1142
153.2411	155.5416	2309



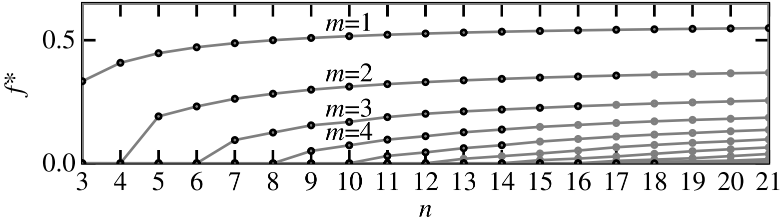
Evolutionary algorithm on SUN Fire E15k: unit simplices, $m = 2,\ t_c = 10$s

	1 processor				8 processors			
n	f^*_{min}	f^*_{mean}	f^*_{max}	$perc$	f^*_{min}	f^*_{mean}	f^*_{max}	$perc$
6	**0.1869**	0.1869	0.1869	100	**0.1869**	0.1869	0.1869	100
7	**0.2247**	0.2247	0.2247	100	**0.2247**	0.2247	0.2247	100
8	**0.2569**	0.2569	0.2569	100	**0.2569**	0.2569	0.2569	100
9	**0.2759**	0.2759	0.2759	100	**0.2759**	0.2759	0.2759	100
10	**0.2936**	0.2936	0.2936	100	**0.2936**	0.2936	0.2936	100
11	**0.3058**	0.3058	0.3058	100	**0.3058**	0.3058	0.3058	100
12	**0.3167**	0.3167	0.3167	100	**0.3167**	0.3167	0.3167	100
13	0.3249	0.3249	0.3249	100	0.3249	0.3249	0.3249	100
14	0.3325	0.3325	0.3330	93	0.3325	0.3325	0.3325	100
15	0.3384	0.3386	0.3391	70	0.3384	0.3384	0.3384	100
16	0.3439	0.3443	0.3448	25	0.3439	0.3439	0.3443	94
17	0.3484	0.3490	0.3497	8	0.3484	0.3486	0.3490	56
18	0.3526	0.3532	0.3538	3	0.3526	0.3529	0.3531	17
19	0.3562	0.3568	0.3575	2	0.3562	0.3565	0.3568	5
20	0.3597	0.3602	0.3607	4	0.3595	0.3599	0.3602	2
21	0.3625	0.3630	0.3636	4	0.3623	0.3627	0.3631	2

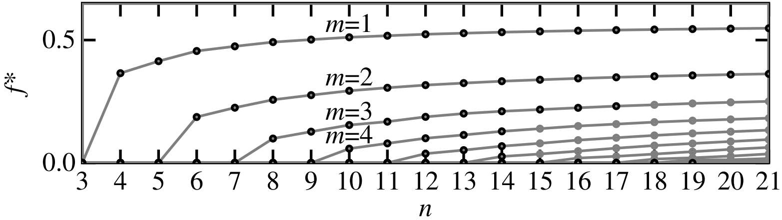
Evolutionary algorithm on SUN Fire E15k: $t_c = 30s/p$

	$p = 1$		$p = 4$		$p = 8$		$p = 12$		$p = 16$	
n	$perc$	f^*	$perc$	f^*	$perc$	f^*	$perc$	f^*	$perc$	f^*
					standard simplices					
5	100	0.1907	100	0.1907	100	0.1907	100	0.1907	100	0.1907
6	100	0.2309	100	0.2309	100	0.2309	100	0.2309	100	0.2309
7	100	0.2621	100	0.2621	100	0.2621	100	0.2621	100	0.2621
8	100	0.2825	100	0.2825	100	0.2825	100	0.2825	100	0.2825
9	100	0.2991	100	0.2991	100	0.2991	100	0.2991	100	0.2991
10	99	0.3115	100	0.3115	100	0.3115	100	0.3115	100	0.3115
11	95	0.3217	100	0.3217	100	0.3217	100	0.3217	100	0.3217
12	79	0.3300	100	0.3300	100	0.3300	100	0.3300	98	0.3300
13	60	0.3371	95	0.3371	86	0.3371	52	0.3371	29	0.3371
14	45	0.3429	87	0.3429	34	0.3429	6	0.3429	1	0.3429
15	35	0.3481	20	0.3481	2	0.3481	1	0.3481	1	0.3484
16	26	0.3525	7	0.3525	1	0.3527	1	0.3527	1	0.3527
					unit simplices					
6	100	0.1869	100	0.1869	100	0.1869	100	0.1869	100	0.1869
7	100	0.2247	100	0.2247	100	0.2247	100	0.2247	100	0.2247
8	100	0.2569	100	0.2569	100	0.2569	100	0.2569	100	0.2569
9	100	0.2759	100	0.2759	100	0.2759	100	0.2759	100	0.2759
10	100	0.2936	100	0.2936	100	0.2936	100	0.2936	100	0.2936
11	100	0.3058	100	0.3058	100	0.3058	100	0.3058	100	0.3058
12	100	0.3167	100	0.3167	100	0.3167	100	0.3167	100	0.3167
13	91	0.3249	100	0.3249	100	0.3249	77	0.3249	62	0.3249
14	92	0.3325	100	0.3325	49	0.3325	20	0.3325	13	0.3325
15	69	0.3384	61	0.3384	4	0.3384	3	0.3384	2	0.3384
16	64	0.3439	8	0.3439	1	0.3439	1	0.3443	3	0.3443
					cubes					
8	100	0.2245	100	0.2245	100	0.2245	100	0.2245	100	0.2245
16	35	0.2965	1	0.2965	1	0.2965	1	0.2974	1	0.3009

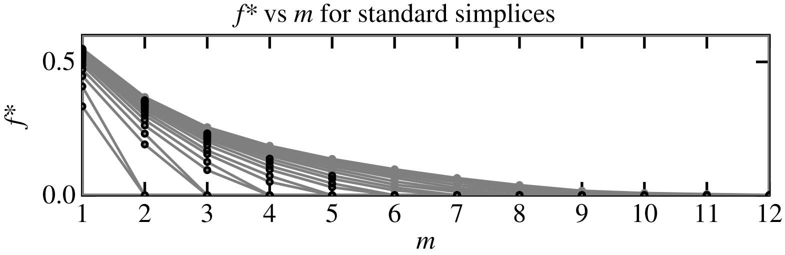
f^* vs n for standard simplices

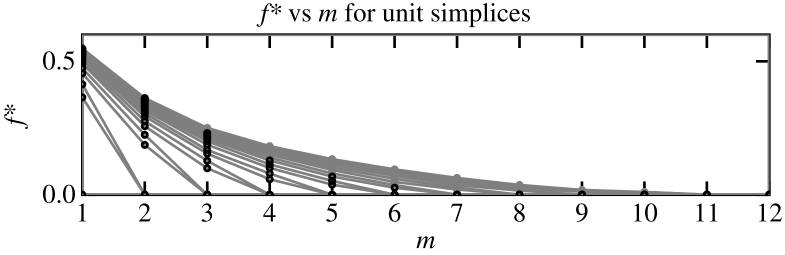


f^* vs n for unit simplices

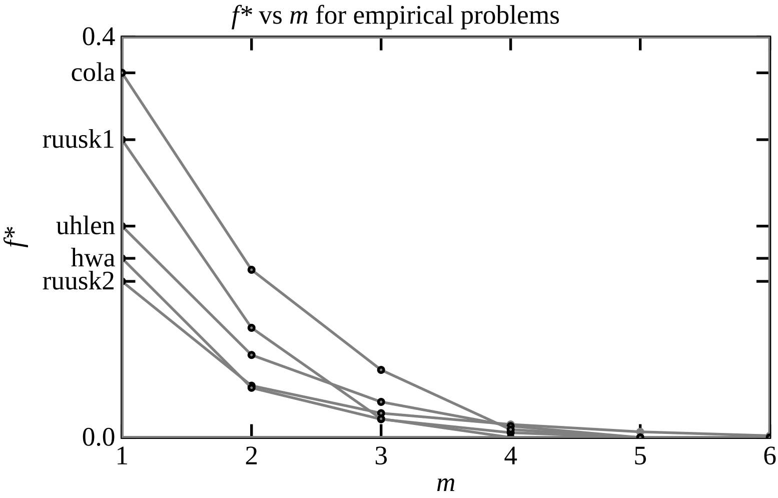


On dimensionality of embedding space

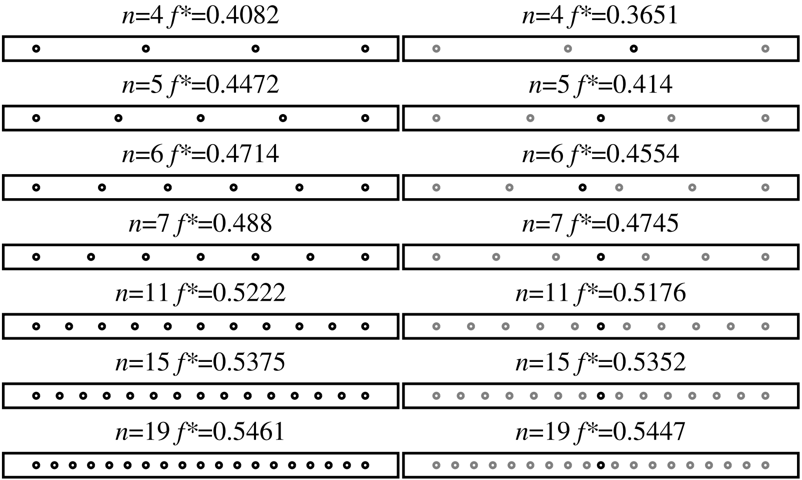




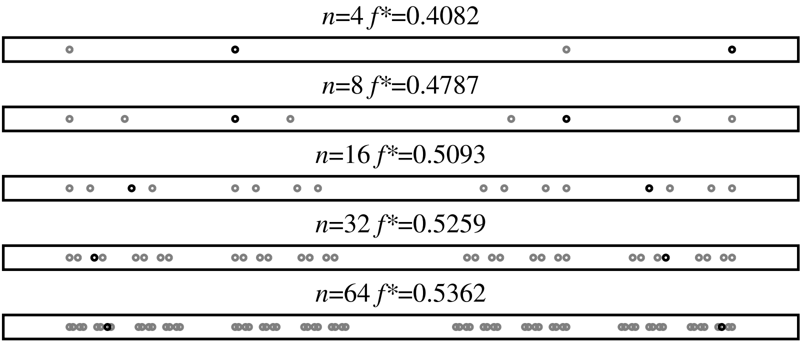
On dimensionality of embedding space



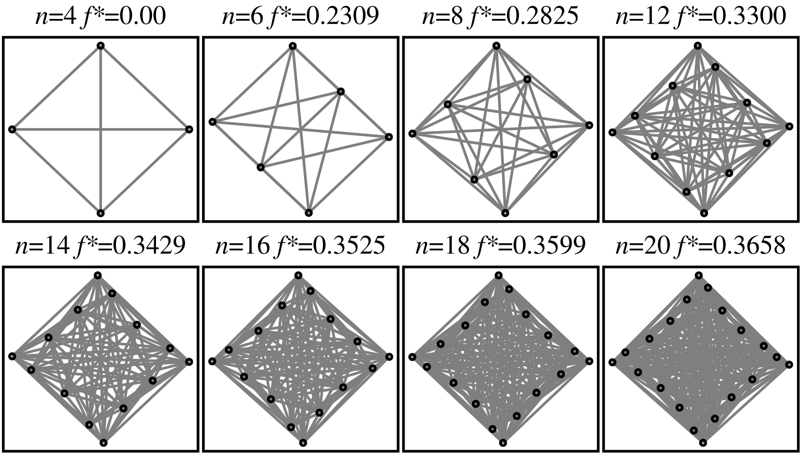
Images of simplices, $m = 1$



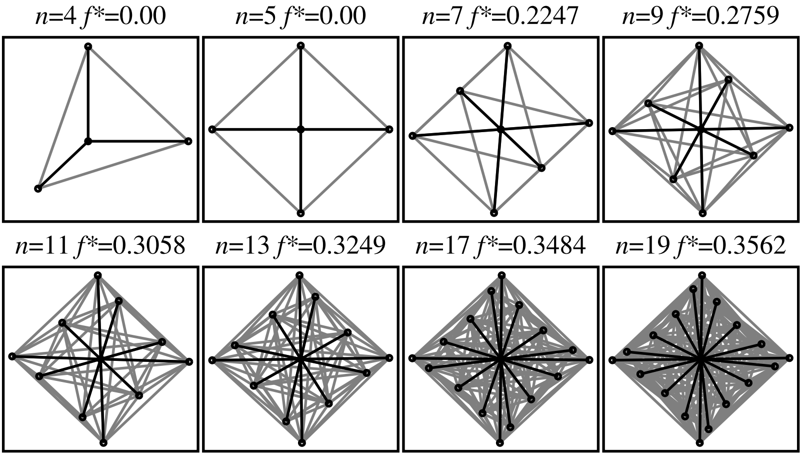
Images of cubes, $m = 1$



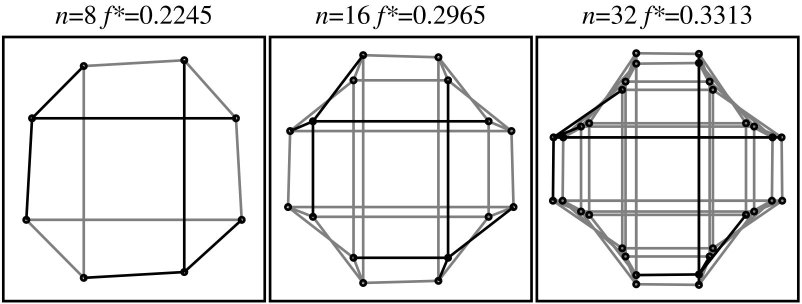
Images of standard simplices, $m = 2$



Images of unit simplices, $m = 2$

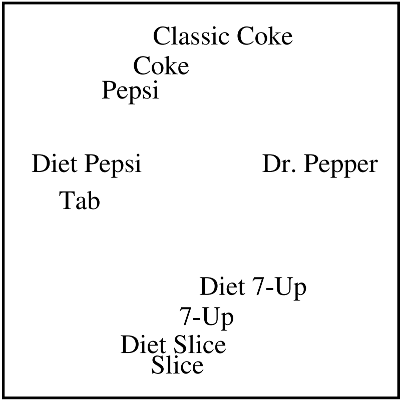


Images of cubes, $m = 2$

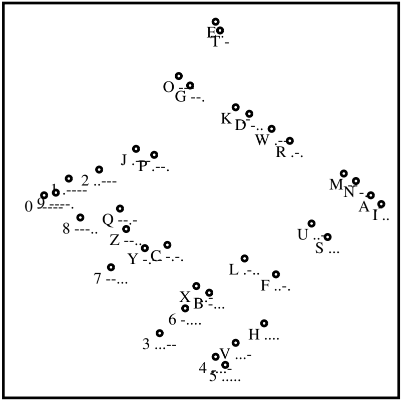


Images of empirical data, $m = 2$

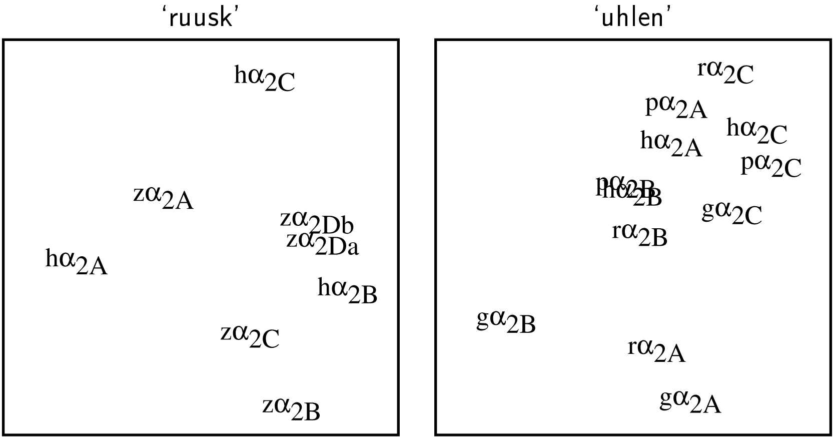
'cola'



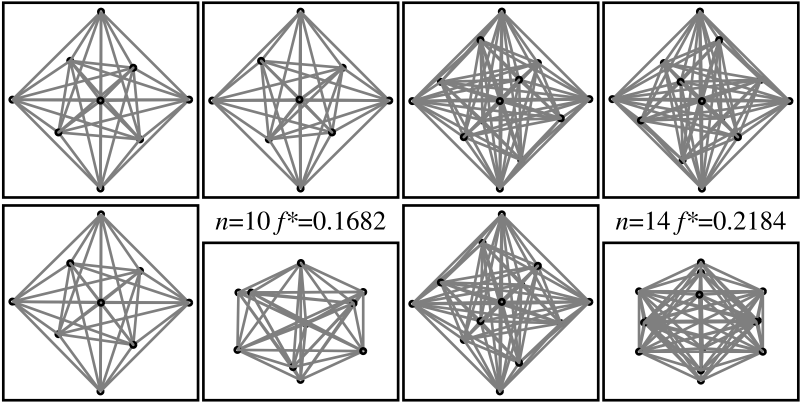
'morseodes'



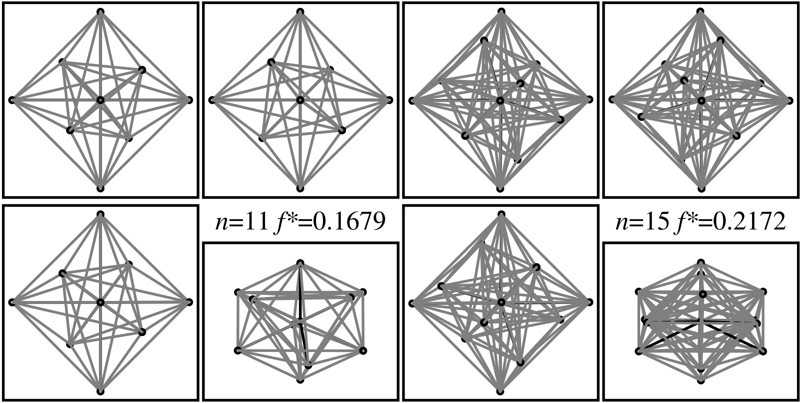
Images of pharmacological data, $m = 2$



Images of standard simplices, $m = 3$



Images of unit simplices, $m = 3$



Images of cubes, $m = 3$

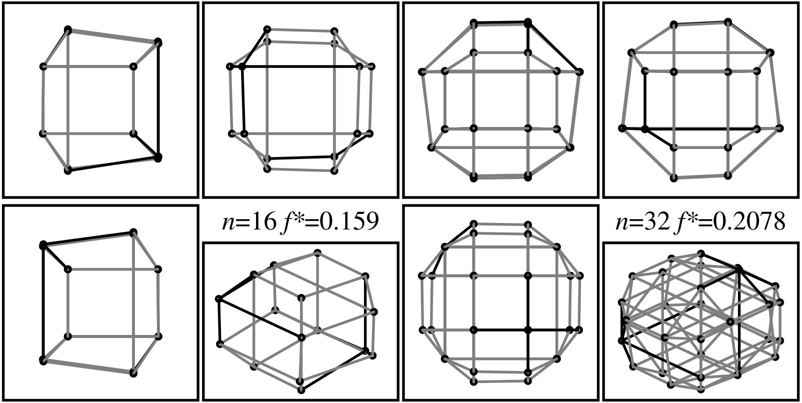


Image of the properties of human and zebrafish α_2-adrenoceptors, $m = 3$

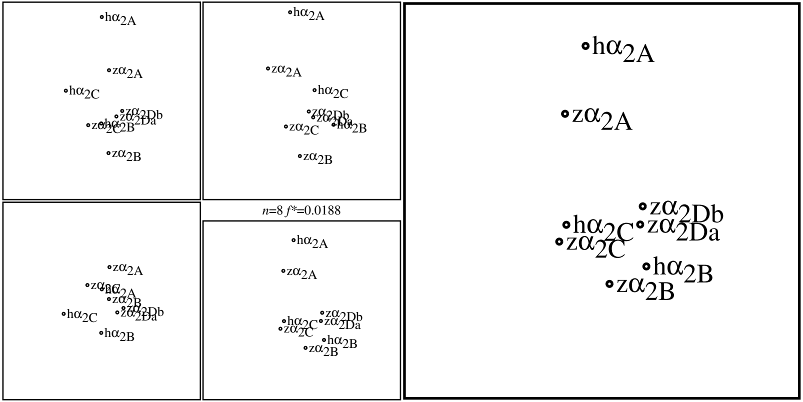


Image of the properties of 20 ligands binding human and zebrafish α_2-adrenoceptors, $m = 3$

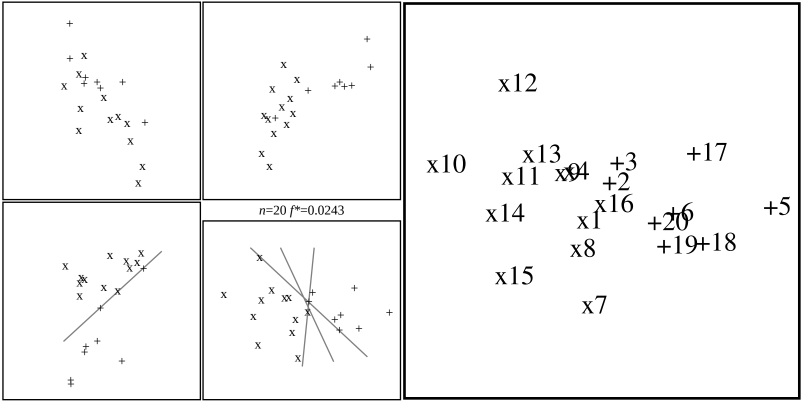


Image of the properties of human, rat, guinea pig and pig α_2-adrenoceptors, $m = 3$

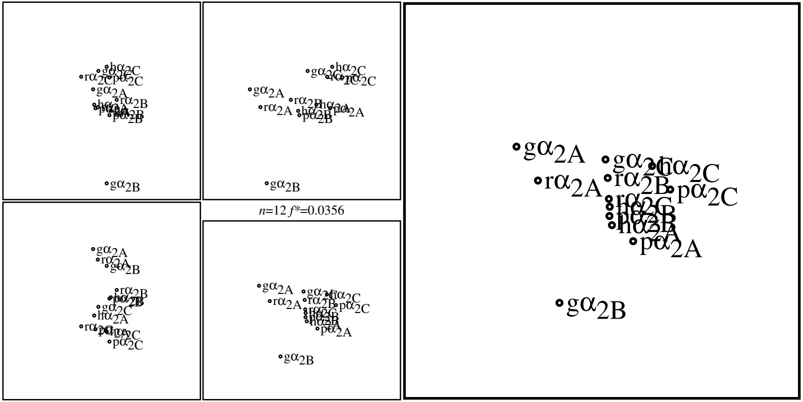
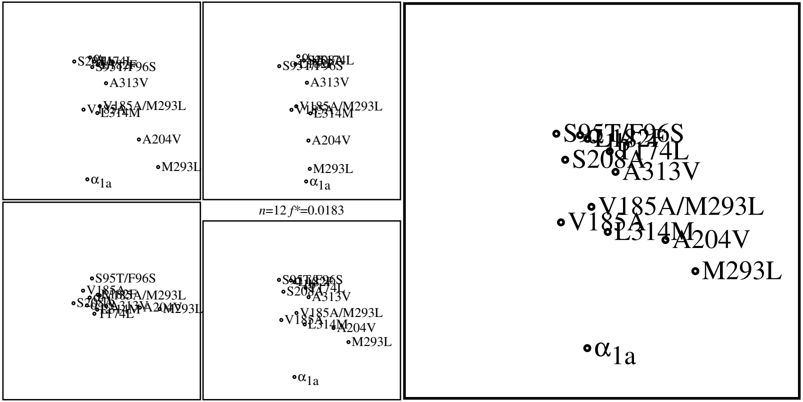


Image of the properties of wild type and mutant α_1-adrenoceptors, $m = 3$



Спасибо за внимание

Thank you for your attention