

Estimation of Distribution Algorithms for Combinatorial Problems in Bioinformatics

Roberto Santana

Intelligent Systems Group
The University of the Basque Country
San Sebastian, Spain
roberto.santana@ehu.eus

(Lyon, October 2, 2025)



UPV - EHU

Index

- TagSNP selection problem
- Estimation of distribution algorithms (EDAs)
- Transfer learning in EDAs
- EDAs for TagSNP selection

TagSNPs Problem

Single Nucleotide Polymorphisms

More than 99.9% sequence of DNA is similar between different populations

SNiPs or SNPs =
sites of variation in the genome
(spelling mistakes)



Karen	AGCTTGAC	TCCA	TGATGATT
Debo	AGCTTGAC	GCCA	TGATGATT
Jose	AGCTTGAC	TCC	CTGATGATT
Thomas	AGCTTGAC	GCC	CTGATGATT
Anupriya	AGCTTGAC	TCCA	TGATGATT
Robert	AGCTTGAC	GCCA	TGATGATT
Michelle	AGCTTGAC	TCC	CTGATGATT
Zhijun	AGCTTGAC	GCC	CTGATGATT

Most of the variations come given by single nucleotide polymorphism (SNPs)

TagSNPs Selection Problem

- If an individual carries out a SNP then, with high probability will carry out another “close” SNP
- The value of one SNP can be used to predict the value of other SNP
- TagSNP selection problem:
Given a set of SNPs, find the minimum subset that can be used to predict the rest of SNPs
- Prediction ability is given in terms of correlation

TagSNPs Selection Problem

- Prediction ability is given in terms of correlation threshold r_{min}
- Single Marker: A SNP s_i tags another SNP s_j if $r_{i,j} > r_{min}$
- Multiple Marker: A subset of SNPs T tags a single SNP s_i , if $r_{T,i} > r_{min}$

We consider marker subsets with $|T| \leq 2$

Santana, R., Mendiburu, A., Zaitlen, N., Eskin, E., and Lozano, J. A. (2010). Multi-marker tagging single nucleotide polymorphism selection using estimation of distribution algorithms. Artificial Intelligence in Medicine, 50(3), 193-201.



TagSNPs Selection Problem

Codification

- $\mathbf{x} = (x_1, \dots, x_n)$ where

$$x_i = \begin{cases} 1 & \text{if } s_i \text{ is a tagSNP} \\ 0 & \text{otherwise} \end{cases}$$

Optimization problem

$$\begin{aligned} \min f(\mathbf{x}) &= \sum_{i=1}^n x_i \\ \text{subject to } \mathbf{x} &\text{ tags all the SNPs} \end{aligned}$$



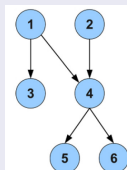
Datasets

<i>Index</i>	<i>Population</i>	<i>n.ind.</i>	<i>SNPs</i>	<i>n'</i>	<i>pairs</i>	<i>triples</i>
1	<i>Adygei</i>	15	1388	1048	3445	203640
2	<i>Balochi</i>	24	1399	1020	3486	220892
3	<i>BantuKenya</i>	11	1338	800	2183	89372
4	<i>BantuSouthAfrica</i>	8	1379	905	2553	127780
5	<i>Basque</i>	24	1362	1042	3956	252296
6	<i>Bedouin</i>	45	1433	1015	3147	182836
7	<i>Bengali</i>	15	1390	1007	3374	211439
8	<i>BiakaPygmy</i>	23	1322	700	2071	78136
9	<i>Brahui</i>	24	1368	975	2770	140452
10	<i>Burusho</i>	23	1388	1016	3272	197048
11	<i>CEU</i>	60	1386	1018	3227	191730
12	<i>Cambodian</i>	8	1335	1067	3476	219632
13	<i>Colombian</i>	7	1175	1008	4467	264906
14	<i>Dai</i>	10	1267	982	3762	212758
15	<i>Daur</i>	10	1248	956	3372	187016
16	<i>Druze</i>	41	1377	1019	3306	190120
17	<i>French</i>	28	1418	1068	3596	229216
18	<i>Han</i>	34	1273	946	3527	211194
19	<i>Han – NChina</i>	10	1230	954	3270	188553
20	<i>Hazara</i>	22	1329	968	2868	159534



Probabilistic graphical model

Graphical models



(a) Structure

X_1	1	1	1	2	2	2
X_2	1	2	3	1	2	3
$P(X_3=1 \mid X_1, X_2)$	θ_{411}	θ_{421}	θ_{431}	θ_{441}	θ_{451}	θ_{461}
$P(X_3=2 \mid X_1, X_2)$	θ_{412}	θ_{422}	θ_{432}	θ_{442}	θ_{452}	θ_{462}
$P(X_3=3 \mid X_1, X_2)$	θ_{413}	θ_{423}	θ_{433}	θ_{443}	θ_{453}	θ_{463}
$P(X_4=1 \mid X_1, X_2)$	θ_{414}	θ_{424}	θ_{434}	θ_{444}	θ_{454}	θ_{464}
$P(X_4=2 \mid X_1, X_2)$	θ_{415}	θ_{425}	θ_{435}	θ_{445}	θ_{455}	θ_{465}

(b) Conditional probability table

A probabilistic graphical model for $\mathbf{X} = (X_1, X_2, \dots, X_n)$ encodes a graphical factorization of a joint probability distribution $p(\mathbf{x})$

- It has two components:
 - A structure $\mathcal{S}$ (e.g. directed acyclic graph for Bayesian networks).
 - A set of local marginal probability values.
- $\mathcal{S}$ represents a set of conditional independence assertions between the variables.



Optimization approaches

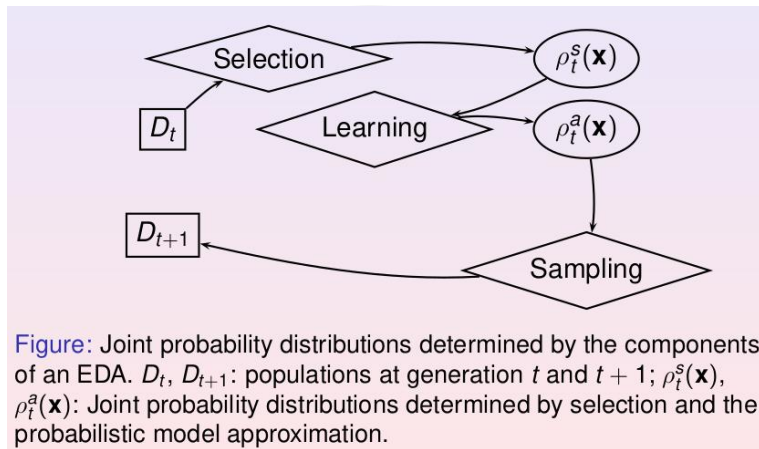
Optimization problem

$$\hat{f} = \max_{x_i \in \{1, \dots, K\}^n} f(\mathbf{x})$$

EDAs

- Population based evolutionary algorithms
- Use probabilistic models instead of genetic operators
- At each generation a probabilistic model of the selected population is learnt
- The probabilistic model is used to sample new solutions

Probabilistic distributions in EDAs



Pseudocode of UMDA

UMDA (Muhlenbein and Paas:1996)

- 1 $D_0 \leftarrow$ Generate M solutions randomly
- 2 $l = 1$
- 3 **do** {
- 4 $D_{l-1}^s \leftarrow$ Select $N \leq M$ solutions from D_{l-1}
according to a selection method
- 5 $p_l(\mathbf{x}) = \prod_{i=1}^n p(x_i | D_{l-1}^s) \leftarrow$ Estimate the joint
probability of the selected solutions
- 6 $D_l \leftarrow$ Sample M solutions (the new population)
from $p_l(\mathbf{x})$
- 7 } **until** A stop criterion is met



EDAs: other probabilistic model used

Markov-chain model

$$p_{MK}(\mathbf{x}) = p(x_1, \dots, x_{k+1}) \prod_{i=k+2}^n p(x_i \mid x_{i-1}, \dots, x_{i-k})$$

Tree model

$$p_{Tree}(\mathbf{x}) = \prod_{i=1}^n p(x_i \mid pa(x_i))$$

Mixture of trees model

$$p_{MT}(\mathbf{x}) = \sum_{j=1}^m \lambda_j p_{Tree}^j(\mathbf{x})$$

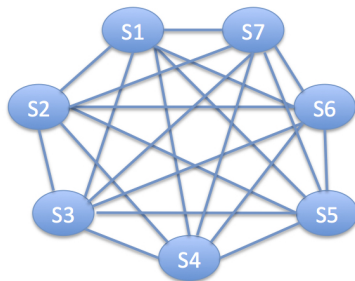
EDA approaches: Tree model

Classical and restricted tree-EDA approaches

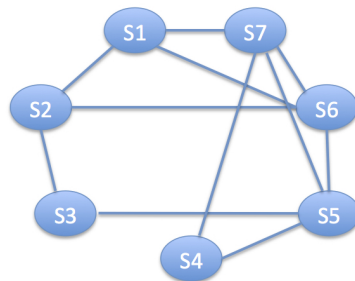
- Tree-EDA: The tree is learnt from the matrix of mutual information learned from data
- Tree-EDA^r: The tree is learnt from the data constrained to known interaction of problem instance



EDA for tagSNP selection problem

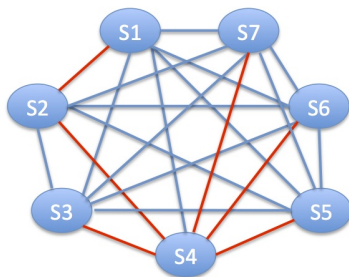


TreeEDA

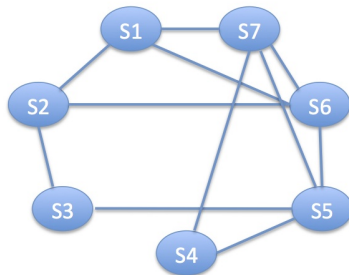


TreeEDA'

EDA for tagSNP selection problem

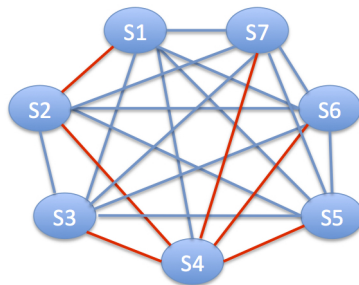


TreeEDA

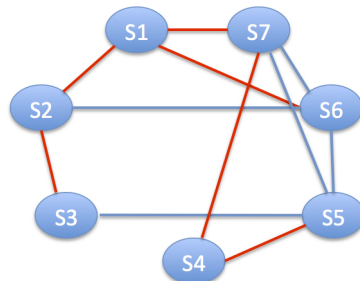


TreeEDA'

EDA for tagSNP selection problem



TreeEDA



TreeEDA^r

Datasets

Characteristics of the datasets

- No. of datasets: 58
- No. of SNPs: 780–1089
- No. of pairs: 1937–4467
- No. of triples: 76007–264906

Algorithm Parameters

- Pop. Size: 5000
- Selection: 15
- Repetitions: 30
- Generations: 100



Preliminary Results

<i>I</i>	<i>Best</i>			<i>mean</i>		
	<i>UMDA</i>	<i>Tree</i>	<i>TreeR</i>	<i>UMDA</i>	<i>Tree</i>	<i>TreeR</i>
1	362	285	285	367.63	291.30	<u>288.13</u>
2	365	302	300	371.17	304.50	<u>303.73</u>
3	344	284	283	349.73	285.97	<u>285.53</u>
4	372	305	306	378.47	308.67	<u>308.33</u>
5	354	283	279	360.80	286.53	<u>284.40</u>
6	376	299	298	381.63	302.37	<u>301.73</u>
7	341	276	273	346.93	281.27	<u>280.00</u>
8	325	265	264	329.23	265.70	<u>265.43</u>
9	364	296	294	374.53	299.37	<u>298.40</u>
10	351	288	288	359.30	<u>291.43</u>	291.50
11	360	293	294	370.00	299.43	<u>297.50</u>
12	341	270	270	346.93	276.00	<u>273.97</u>
13	286	235	233	294.20	238.90	<u>236.30</u>
14	325	263	261	332.37	266.90	<u>263.13</u>
15	325	264	264	332.90	<u>267.63</u>	267.67
16	361	291	291	369.90	297.37	<u>296.50</u>
17	360	284	284	374.17	290.70	<u>289.07</u>

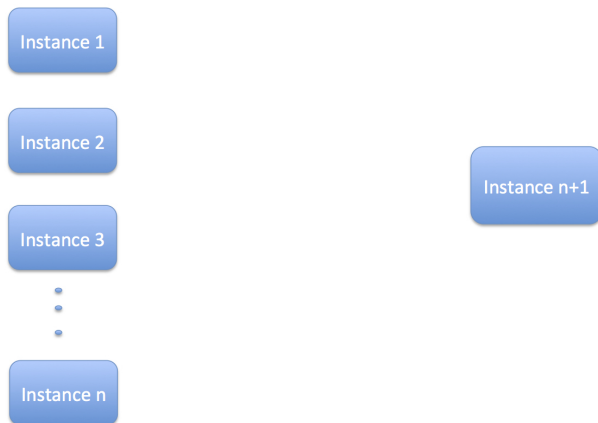


Results

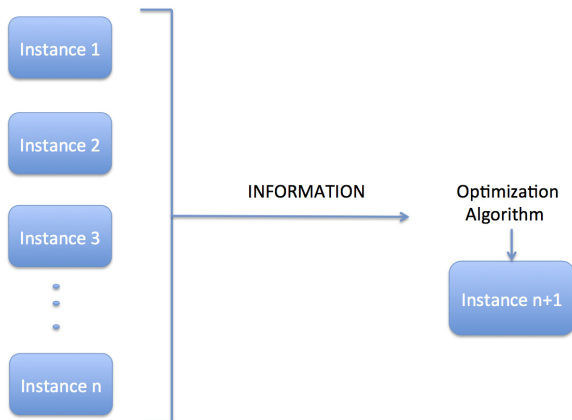
TreeEDA vs TreeEDA^r

- Average results: TreeEDA^r obtains better results in 43 from 58 datasets
- Best results: TreeEDA^r obtains better results in 26 from 58 datasets and the same results in 15 of them

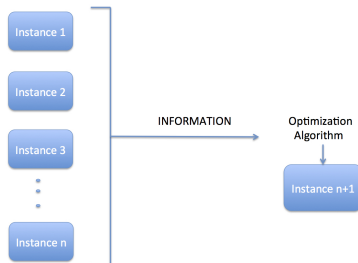
Solving new instances



Solving new instances



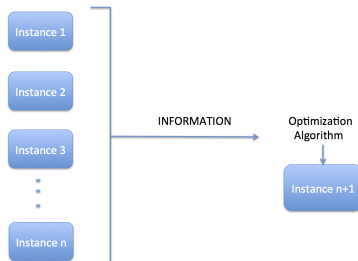
Transfer Learning



Questions

- What information extract from the problems?
- How to define relatedness between problems?
- How to transfer the available information?

Transfer Learning



Transfer Learning for Optimization

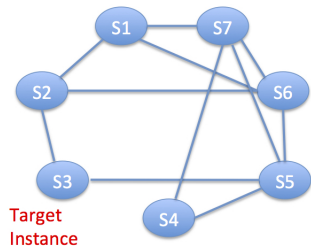
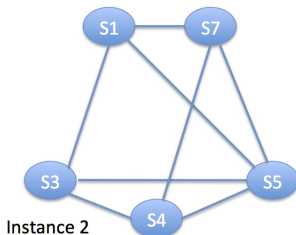
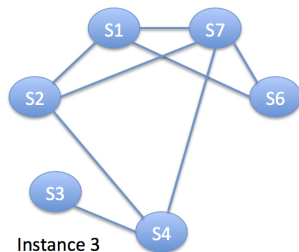
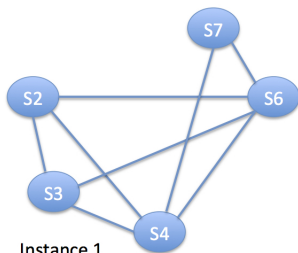
- **Transfer of solutions**
- Structural transfer
- Behavioral or algorithmic transfer
- Combined transfer

Transfer Learning in the TagSNP problem

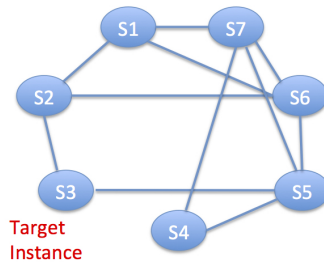
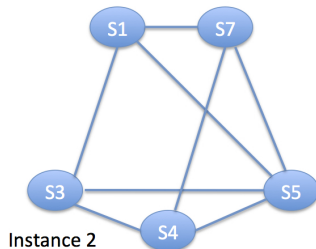
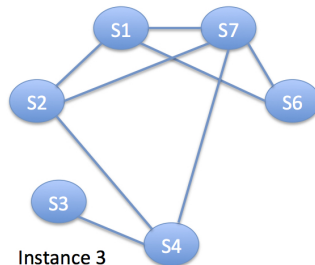
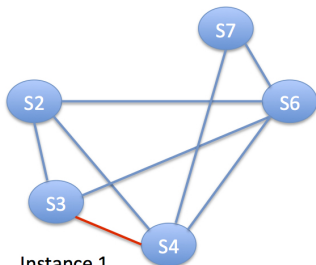
Structural Transfer

- The **possible dependencies** between the variables are transferred
- Four different transfer strategies are evaluated

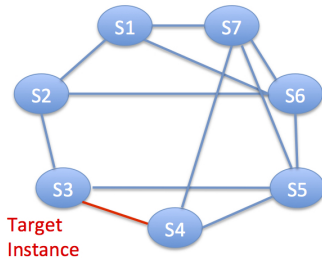
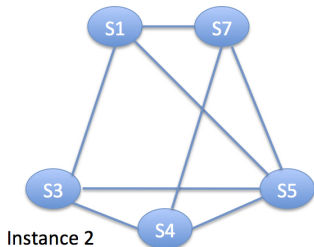
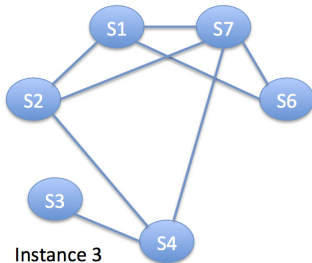
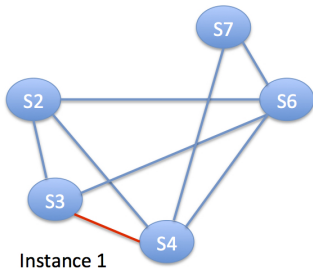
Tn1: Aggregated Interactions



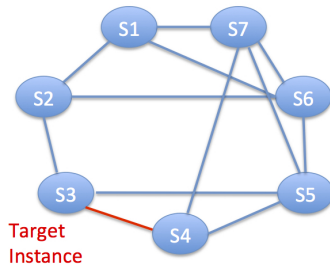
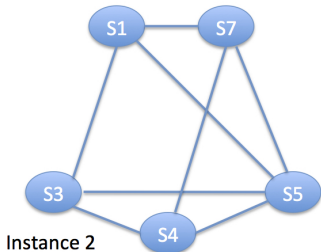
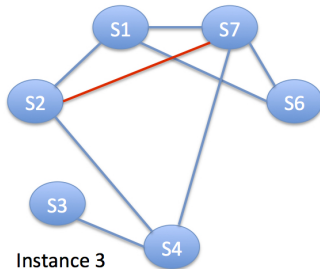
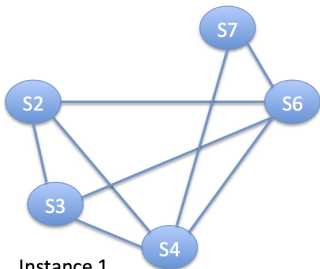
Tn1: Aggregated Interactions



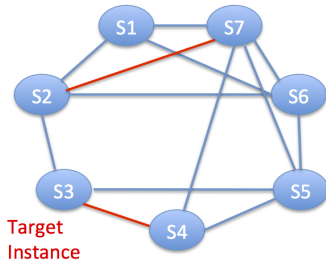
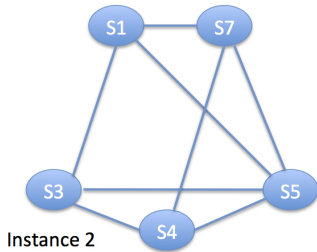
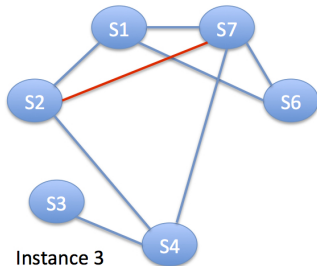
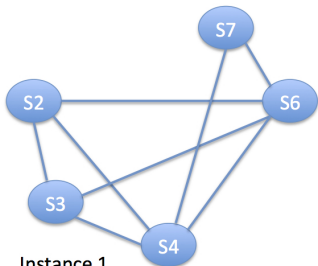
Tn1: Aggregated Interactions



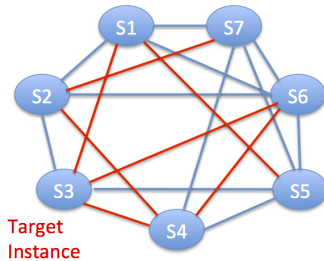
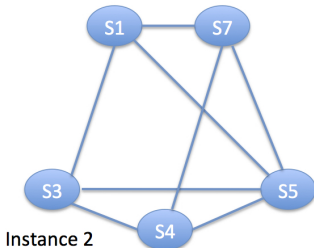
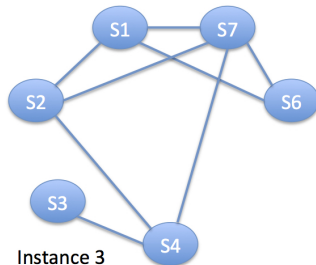
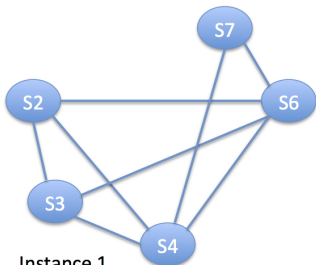
Tn1: Aggregated Interactions



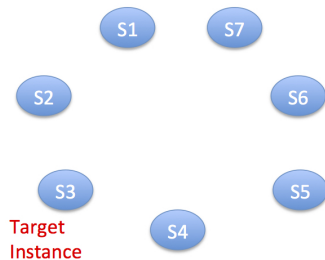
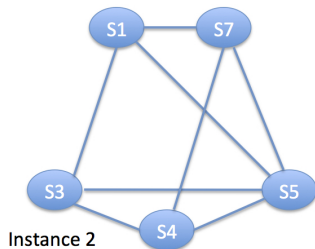
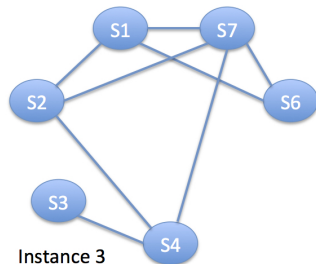
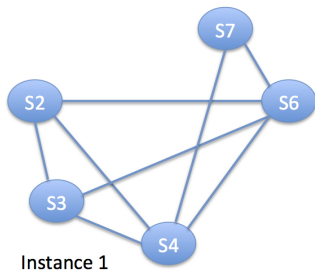
Tn1: Aggregated Interactions



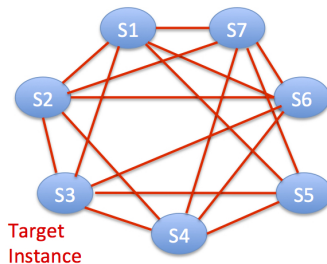
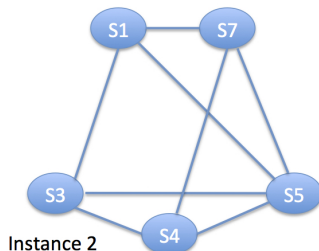
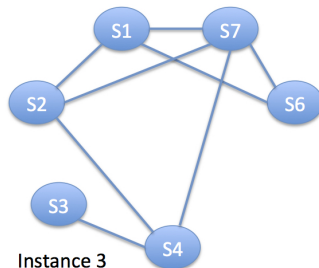
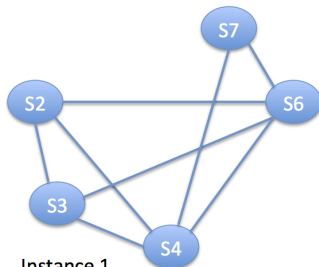
Tn1: Aggregated Interactions



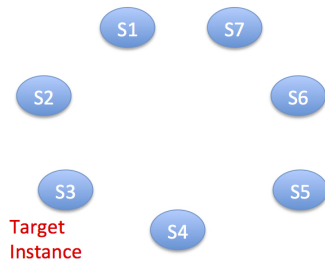
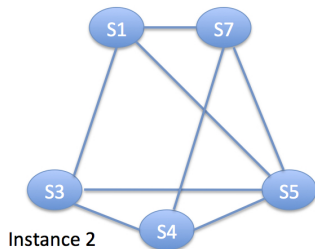
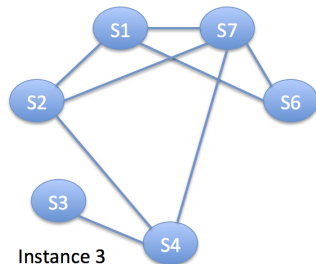
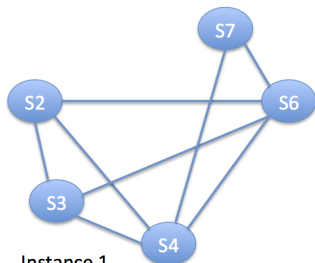
Tn2: Pure Transfer



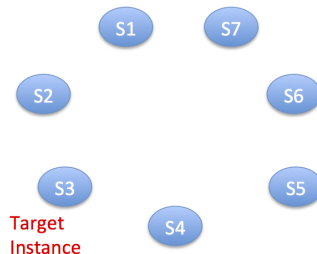
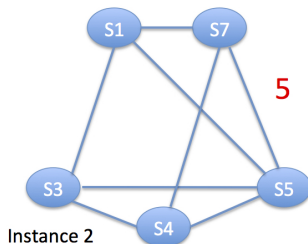
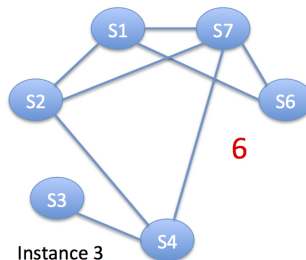
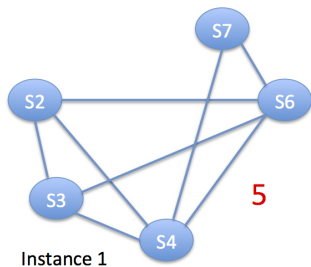
Tn2: Pure Transfer



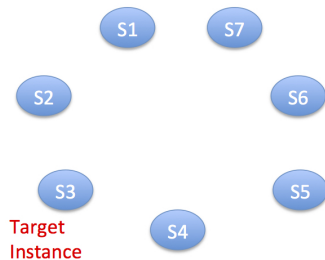
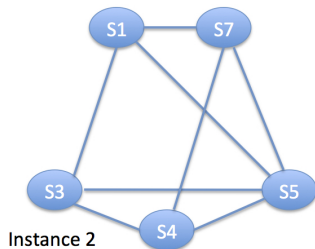
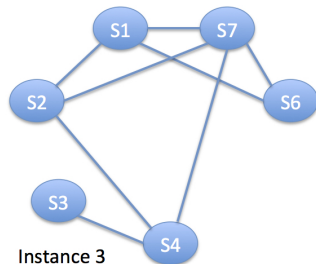
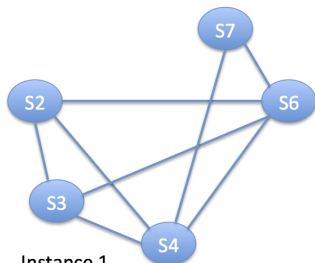
Tn3: Limited Information. Overlapping Variables



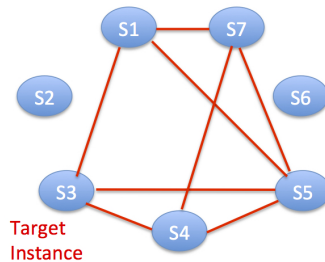
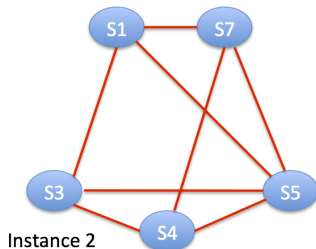
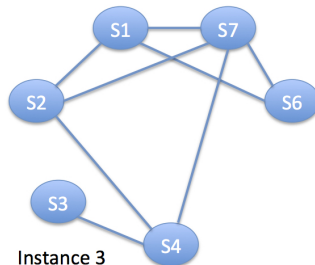
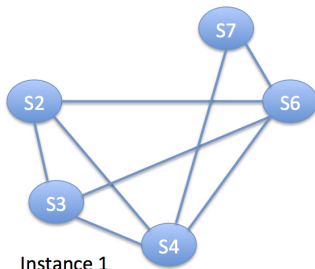
Tn3: Limited Information. Overlapping Variables



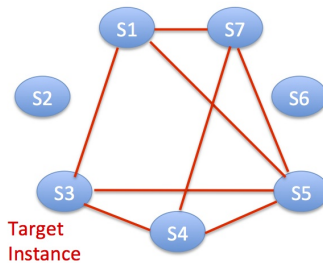
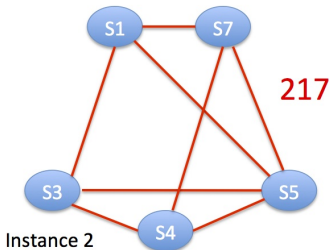
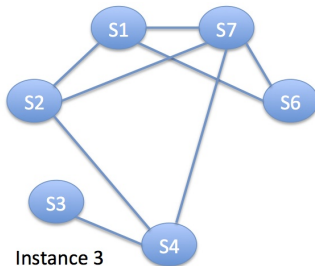
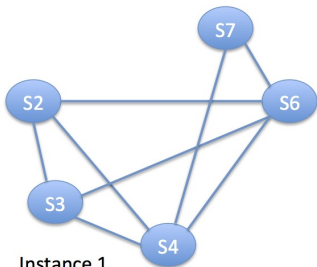
Tn4: Limited Information. EDA results



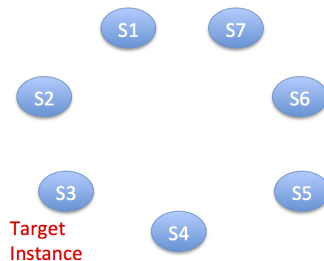
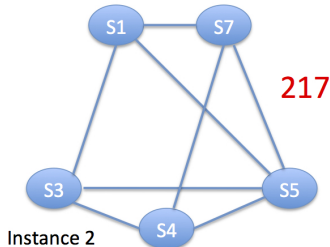
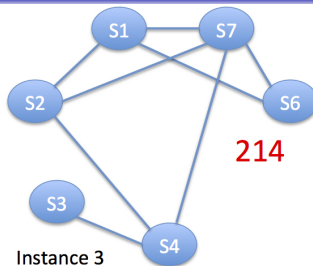
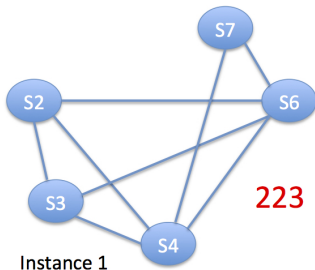
Tn4: Limited Information. EDA results



Tn4: Limited Information. EDA results



Tn4: Limited Information. EDA results



Transfer Learning in the tagSNP Problem. Tn1 and Tn2

Tn1

- A matrix of interactions for problem i is created by combining the interactions between any pair of SNPs in problem i that were detected in any of the other SNP datasets

Tn2

- Identical to Tn1 but information about the interactions in problem i is not included in the aggregation matrix.

Transfer Learning in tagSNP Problem Tn3 and Tn4

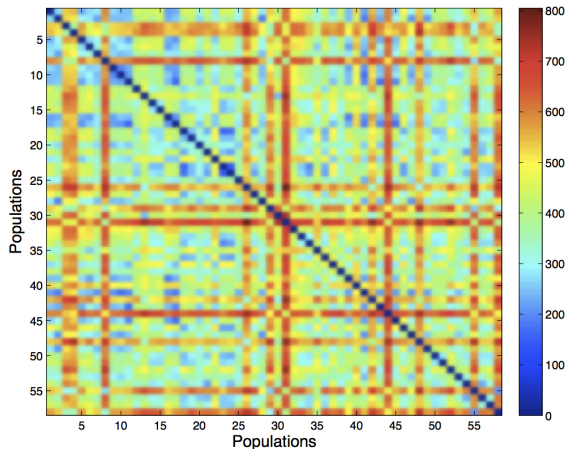
Tn3

- The most related problems to problem i are selected as those that share the largest set of common variables (SNPs) with it.
- The aggregation matrix is formed using the matrix of interactions from this subset of problems.

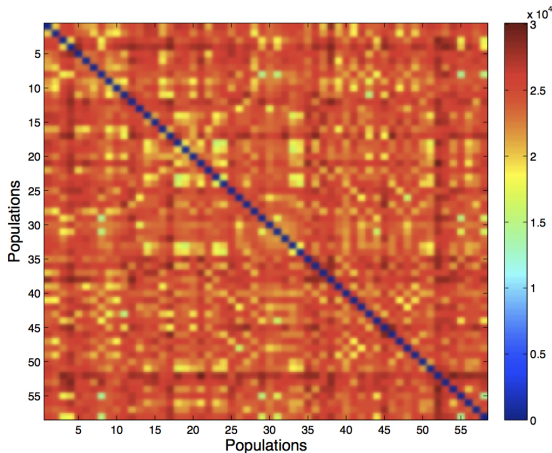
Tn4

- The interaction matrix of each problem j are used to solve problem i
- The best interaction matrices for problem i are aggregated

Similarities between problems (share SNPs)



Similarities between problems (structural similarity)



Transfer Learning Results

<i>mean</i>			<i>I</i>	<i>Best</i>				<i>mean</i>			
<i>UMDA</i>	<i>Tree</i>	<i>TreeR</i>		<i>Tn1</i>	<i>Tn2</i>	<i>Tn3</i>	<i>Tn4</i>	<i>Tn1</i>	<i>Tn2</i>	<i>Tn3</i>	<i>Tn4</i>
367.63	291.30	<u>288.13</u>	1	281	281	280	283	285.60	<u>285.20</u>	287.00	287.33
371.17	304.50	<u>303.73</u>	2	296	298	299	297	<u>300.13</u>	300.97	302.93	302.00
349.73	285.97	<u>285.53</u>	3	281	282	283	284	<u>284.27</u>	284.60	285.80	285.13
378.47	308.67	<u>308.33</u>	4	304	304	306	306	306.47	<u>306.17</u>	308.60	309.03
360.80	286.53	<u>284.40</u>	5	276	278	282	282	281.53	<u>281.47</u>	286.93	286.00
381.63	302.37	<u>301.73</u>	6	295	297	295	294	299.23	<u>298.97</u>	300.50	299.00
346.93	281.27	<u>280.00</u>	7	271	269	271	271	275.60	<u>275.00</u>	277.97	275.80
329.23	265.70	<u>265.43</u>	8	263	263	265	264	265.07	<u>264.63</u>	266.43	265.73
374.53	299.37	<u>298.40</u>	9	291	293	291	292	<u>294.80</u>	295.73	295.47	296.17
359.30	<u>291.43</u>	291.50	10	284	285	283	284	<u>287.33</u>	288.20	287.80	287.80
370.00	299.43	<u>297.50</u>	11	291	291	289	288	296.50	295.67	295.33	<u>294.23</u>
346.93	276.00	<u>273.97</u>	12	265	266	270	269	<u>271.07</u>	271.70	274.67	274.20
294.20	238.90	<u>236.30</u>	13	230	229	229	232	233.57	<u>233.47</u>	235.83	236.80
332.37	266.90	<u>263.13</u>	14	257	255	260	258	<u>261.03</u>	261.27	263.37	262.83
332.90	<u>267.63</u>	267.67	15	260	261	266	261	264.73	<u>264.47</u>	268.17	264.97
369.90	297.37	<u>296.50</u>	16	290	286	289	288	293.63	<u>292.17</u>	292.40	293.60
374.17	290.70	<u>289.07</u>	17	281	279	282	284	<u>285.80</u>	286.10	287.27	288.63
331.87	262.37	<u>261.50</u>	18	255	255	255	256	<u>258.13</u>	258.43	258.63	259.17
336.60	269.43	<u>268.00</u>	19	261	262	260	262	<u>265.90</u>	266.30	266.03	267.00
366.77	296.37	<u>294.00</u>	20	287	288	289	291	292.40	<u>292.03</u>	293.33	295.03
332.33	263.27	<u>260.70</u>	21	253	254	254	253	<u>257.27</u>	257.27	258.77	257.67
340.97	278.23	<u>277.27</u>	22	270	270	272	271	274.53	274.67	275.80	<u>274.47</u>
327.50	261.23	<u>260.00</u>	23	253	254	255	254	257.10	<u>256.43</u>	257.93	257.00
360.00	282.43	<u>280.80</u>	24	271	272	274	272	277.37	<u>277.20</u>	278.97	277.33



Transfer Learning Results

- Tn1 improves all the results of TreeEDA and TreeEDA'
- Tn2 improves in all but one
- Tn3 improves in 91% and 79% TreeEDA and TreeEDA' respectively
- Tn4 improves in 90% and 76% TreeEDA and TreeEDA' respectively

Conclusions

- Transfer learning is an attractive alternative for EAs and EDAs
- Simple transfer learning can dramatically improve the results of optimization algorithms
- It is a new area in optimization that deserves much attention



Estimation of Distribution Algorithms for Combinatorial Problems in Bioinformatics

Roberto Santana

Intelligent Systems Group
The University of the Basque Country
San Sebastian, Spain
roberto.santana@ehu.eus

(Lyon, October 2, 2025)



UPV - EHU

Protein problems addressed

Protein folding in HP model

- To find the HP configuration with lowest energy

Side chain placement problem

- To find the optimal positioning of the side chain with respect to a given backbone

Protein design

- To find the sequence that has the lowest energy with respect to a protein structure



HP and functional model protein

Example

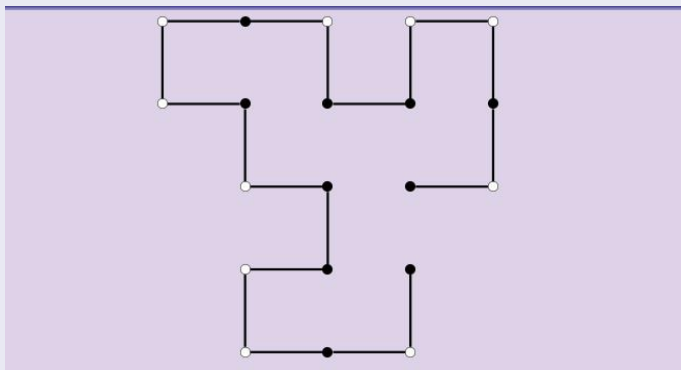


Figure: Best solution found for sequence
HPHPPHHPHPPHHPHPPH

Results in the two-dimensional lattice

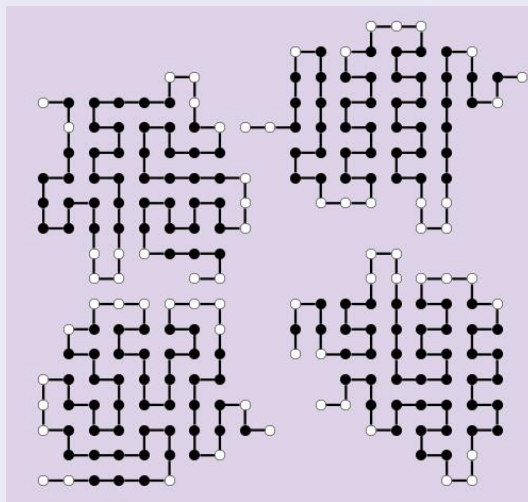


Figure: Optimal solution and three sub-optimal solutions for the s7