ET36/UEFISCSU: Investigatii Structurale Integrate pe Compusi Biologic Activi (MEC /3150/11.10.2005) - Raport de cercetare (în extenso)

Lorentz JÄNTSCHI, Sorana D. BOLBOACĂ

Cadru:

Programul:	Cercetare de Excelenţă
Modul: II	Proiecte de Dezvoltare a Resurselor Umane pentru Cercetare
Tipul proiectului:	Proiecte de cercetare de excelenta pentru tinerii cercetători
Cod proiect:	ET36/2005
Denumirea proiectului:	Investigaţii Structurale Integrate pe Compuşi Biologic Activi

Amplasament:

Ministerul Educaţiei, Cercetării şi Tineretului
Universitatea Tehnică din Cluj-Napoca
Facultatea de Ştiinţa şi Ingineria Materialelor
Catedra de Chimie
C501, B-dul Muncii 103-105, 400641 Cluj-Napoca, Romania, EU

Cuprins

Cadrul general al proiectului

Proiectul intitulat *Investigații Structurale Integrate pe Compuşi Biologic Activi* a fost finanțat din fonduri publice în cadrul programului *Cercetare de Excelență* (acronim: CEEx) aprobat prin H.G. 368/28.04.2005 la inițiativa Ministerului Educației şi Cercetării în cadrul modulului II (acronim: M2): *Proiecte de dezvoltare a resurselor umane pentru cercetare*, fiind proiect de tipul II: *Proiecte de cercetare pentru tinerii cercetători*, subtipul I: *Proiecte de cercetare de excelență pentru tinerii cercetători* (acronim: ET).

Scopul general al proiectelor ET a fost *încurajarea excelenței şi originalității în creația ştiințifică a tinerilor cercetători, doctori în ştiințe şi doctoranzi din România*. Obiectivele generale ale proiectelor ET au fost: *creşterea atractivității continuării în țară a activității de cercetare de excelență*; *susținerea unor echipe de cercetare cu potențial de excelență ştiințifică la nivel internațional*; *creşterea producției ştiințifice de valoare internațională*. Activitățile specifice proiectelor ET au fost: *cercetări fundamentale şi aplicative cu grad ridicat de originalitate*; *mobilități în sprijinul participării cercetătorilor la manifestări ştiințifice internaționale de mare vizibilitate*; *dobândirea de competențe complementare necesare membrilor din echipa de cercetare într-un laborator de cercetare recunoscut din institute de cercetare sau agenți economici*; *dezvoltarea infrastructurii de cercetare*. Criteriile de eligibilitate pentru proiectele ET au fost pentru *directorul de proiect (doctor în ştiințe; angajat în universități sau institute de cercetare; vârsta maximă de 35 de ani)* şi pentru *echipa de cercetare (doctor în ştiințe sau doctorand; vârsta maximă de 35 de ani)*. Durata proiectelor ET a fost de *24 luni*. Nivelul de finanțare a fost de *maxim 40000 euro, echivalentul în lei*.

Prevederi contractuale specifice

Proiectele ET s-au finanțat prin intermediul unui *Contract de Finanțare* încheiat între persoane juridice *Autoritatea Contractantă* (UEFISCSU) şi *Contractor* (universitatea sau institutul de cercetare în care a fost încadrat directorul de proiect, în cazul de față *Universitatea Tehnică din Cluj-Napoca* - acronim: UTCN). Lucrările efectuate în cadrul proiectului pe parcursul desfăşurării acestuia au fost reglementate prin întocmirea de documente specifice, pe etape (*Raport intermediar de efectuare lucrări, Proces verbal de avizare internă a rezultatelor, State de plată*, şi *Deviz cadru postcalcul*) şi anual (*Sinteza lucrării, Proces verbal de avizare internă a rezultatelor, State de plată, Deviz cadru postcalcul* şi *Lucrare în extenso*).

O prevedere contractuală specifică a constituit-o *încărcarea în formatul web al proiectului de cercetare, pus la dispoziție de UEFISCSU, documentele de contractare, monitorizare şi raportare*.

Cadrul specific al proiectului

Scopul proiectului a fost *realizarea de investigații structurale integrate pe compuşi biologic activi care să ducă la sintetizarea de noi compuşi chimici biologic activi, cu performanțe superioare şi/sau costuri mai mici de producție care să servească la tratamentul medical aplicat oamenilor şi/sau animalelor, obținerea de noi soiuri de plante transgenice ăi mai bună conservare a mediului ambiant*.

Obiectivele cercetării au fost: *Elaborarea modelului de implementare soft şi interfață online*; *Culegere de date*; *Elaborarea de modele*; *Implementarea modulelor de calcul, prelucrare şi afişare a datelor*; *Definitivarea prototipurilor aplicațiilor*; *Exploatarea sistemului realizat*.

Activitățile desfăşurate pentru atingerea obiectivelor au fost: *Elaborare metodologie integrarea aplicațiilor complexe, Participări la manifestări ştiințifice şi dobândirea de competențe complementare*; *Achiziție, instalare, testare şi configurare aparatură suport* (pentru *Elaborarea modelului de implementare soft şi interfață online*), *Efectuarea de experimente cantitative, Măsurări, achiziție şi managementul datelor, Participări la manifestări ştiințifice şi dobândirea de competențe complementare* (pentru *Culegere de date*), *Elaborare metodologie modele, Participări la manifestări ştiințifice şi dobândirea de competențe complementare, Validare modele* (pentru *Elaborarea de modele*), *Proiectare aplicații, Implementare şi testare module, Participări la manifestări ştiințifice şi dobândirea de competențe complementare* (pentru *Implementarea modulelor de calcul, prelucrare şi afişare a datelor*), *Integrare module, testare aplicații, Evaluare sistem integrat şi elaboarare platforma de operare, Specificare şi publicare pe site aplicații* (pentru *Definitivarea prototipurilor aplicațiilor*), *Efectuarea de experimente cantitative, Management de date şi interpretare rezultate, Participări la manifestări ştiințifice şi dobândirea de competențe complementare* (pentru *Exploatarea sistemului realizat*).

Participanții activi la derularea activităților proiectului au fost: Lorentz JÄNTSCHI (n. 1973, dr. în Chimie Organică din 2000), în calitate de investigator principal; Sorana D. BOLBOACĂ (n. 1973, drd. în Informatică Medicală din 2001, dr. în Informatică Medicală din 2006), în calitate de co-investigator; Mihaela L. UNGUREȘAN (n. 1972, drd. în Chimie Fizică din 2000, dr. în Chimie Fizică din 2007), în calitate de co-investigator; Camelia C. AVRAM (n. 1977, drd. în Sisteme Automate din 2001).

Activitățile efectuate s-au încadrat în obiectivele generale ale proiectelor ET astfel:

- cercetări fundamentale şi aplicative cu grad ridicat de originalitate: Proiectarea şi realizarea de modele, Proiectarea, implementarea şi testarea de module, Integrare module şi testare aplicații, Efectuarea de experimente cantitative;

- participarea la manifestări ştiinţifice internaţionale de mare vizibilitate (acronim: MSIMV) şi dobândirea de competenţe complementare (acronim: DCC):
  1. *Biomedical and Health Informatics from Foundations to Applications to Policy*, American Informatics Medical Association, 22-26.Oct.2005, Washington DC - MSIMV;
  2. *2005 Mid-Atlantic Bio: The Conference for Industry and Investors*, MdBio, MAVA, Virginia Biotechnology Association, Technology Council of Maryland, 26-27.Oct.2005, Washington DC - DCC;
  3. *European Federation for Medical Informatics Special Topic Conference - Integrating Biomedical Information: From eCell to ePatient*, European Federation for Medical Information, Romanian Society of Medical Informatics, 6-8.Apr.2006, Timişoara, Romania - MSIMV;
  4. *SizeMat: Workshop on Size-Dependent Effects in Materials for Environmental Protection and Energy Application, Specific Support Action, FP6: EC-INCO-CT-2005-016414*, Institute for General and Innorganic Chemistry, Bulgarian Academy of Sciences, 25-27.Mai.2006, Varna, Bulgaria - DCC;
  5. *Third Humboldt Conference on Computational Chemistry*, University of Sofia, Humboldt, 24-28.Iun.2006, Varna, Bulgaria - MSIMV;
  6. *XXIII International Biometric Conference*, International Biometric Society, McGill University Montreal, 16-21.Iul.2006, Montreal, Canada - DCC;
  7. *1-st European Chemistry Congress*, European Association for Chemical and Molecular Sciences, 27-31.Aug.2006, Budapest, Hungary - MSIMV;
  8. *ISCB27: 27-th Annual Conference of the International Society for Clinical Biostatistics*, International Society for Clinical Biostatistics, 27-31.Aug.2006, Geneva, Switzerland - DCC;
  9. *10-th World Multi-Conference on Systemics, Cybernetics and Informatics*, International Institute of Informatics and Systemics, 16-19.Iul.2006, Orlando, FL USA - MSIMV;
  10. *6-th European Conference on Computational Chemistry*, Working Party for Computational Chemistry, Comenius University Bratislava, Slovak Chemical Society, European Association for Chemical and Molecular Sciences, 3-7.Sept.2006, Bratislava, Slovakia - DCC;
  11. *MetEcoMat: Workshop on Ecomaterials and Processes: Characterization and Metrology, Specific Support Action, FP6: EC-INCO-CT-2005-016414*, Institute of General and Inorganic Chemistry, Bulgarian Academy of Sciences, 19-21.Apr.2007, Plovdiv, Bulgaria - MSIMV;
  12. *ECCC11: The 11th Electronic Computational Chemistry Conference*, Monmouth University New Jersey USA, 2-30.Apr.2007, online www, Internet - DCC;

13. *Math/Chem/Comp 2007 - The 22nd International Course & Conference on the Interfaces among Mathematics, Chemistry & Computer Sciences*, Inter-University Center Dubrovnik, Rudjer Boskovic Institute Zagreb, University of Zagreb, University of Split, International Society for Mathematical Chemistry, International Society for Theoretical Chemical Physics, 11-16.Iun.2007, Dubrovnik, Croatia - MSIMV;
14. *Fourth International Conference of Applied Mathematics and Computing*, University of Chemical Technology and Metallurgy Sofia, Technical University of Plovdiv, 12-18.Aug.2007, Plovdiv, Bulgaria - DCC.

Problematica cercetării

In ultima perioadă de timp, indicii structurali folosiți pentru modelarea de compuşi biologic activi prin studii integrate structură-activitate (SAR - structure activity relationship) sunt tot mai frecvent calculați din parametrii sterici (geometrici) şi/sau electrostatici (sarcini parțiale) [1] în detrimentul parametrilor pur topologici [2]. Sunt preferate calcule semiempirice şi cuantice cu programe ca Hondo95, Gaussian94, Gamess, Icon08, Tx90, Polyrate, Unichem/Dgauss, Allinger`s MM3, Mopac93, Mozyme si HyperChem [3].

Analiza de regresie structură/activitate foloseşte metode clasice ca regresia liniară, liniară multiplă, neliniară, sisteme expert sau rețele neuronale petru baze mari de date [4].

Elaborarea, implementarea şi folosirea modelului molecular pentru obținerea de noi compuşi biologic activi este referită frecvent în fluxul de informatii. Iată un rezumat al acestor preocupări.

Ca metodă preliminară de analiză, unii autori aliniază setul de molecule. Mai mult, metoda CoMFA introduce un algoritm în 6 paşi pentru analiza structură-activitate [5]:

1. construieşte setul şcoală de molecule cu activitate biologică cunoscută şi generează structura 3D a moleculelor (folosind programe ca: Mopac, Sybyl [6-7], HyperChem [8-9], Alchemy2000 [6], MolConn [10];
2. alege o metodă de suprapunere (care poate fi suprapunerea de fragmente alese din molecule [5], [11,12] sau suprapunerea de grupuri farmacofore [13] şi suprapune virtual coordonatele spațiale;
3. construieşte un grid care inconjoară moleculele suprapuse la pasul (2) într-o formă standard sau modificată ([14]) şi alege un atom de probă pentru interacția cu punctele gridului [15,16];
4. utilizează o metodă empirică (Hint [17]), un model specific (suprapunere farmacoforă [18]), energia potențială clasică (Lennard-Jones, Coulomb), energia legaturilor de hidrogen [19], câmpul generat de orbitalii moleculari [20,21] sau orice alt model definit de utilizator [15] şi calculează valorile de interacțiune în gridul de la pasul (3) a câmpului de interacțiune ales cu atomul probă de la pasul (3);
5. foloseşte valorile calculate ale interacțiunii de la pasul (4) între punctele gridului şi atomul de probă pentru a face estimarea activității cunoscute construind o relație structură activitate;
6. foloseşte relația structură activitate obținută la pasul (5) pentru a face predicția activității pentru molecule cu aceeaşi suprapunere cu setul şcoală de la pasul (1).

Metoda CoMFA este un bun instrument pentru predicția unui variat tip de activități biologice cum sunt citotoxicitatea [22], inhibiția [16,21], sau proprietățile de formare [23,24]. Mai

mult, metoda serveşte în modelarea compuşilor cu efecte farmaceutice [13,25] şi inhibitorilor HIV [26].

Revenind asupra investigatiilor structurale pe compuşi biologic activi, o activitate foarte importantă este căutarea substructurilor active biologic din compuşii biologic activi care produc cea mai mare parte a răspunsului biologic măsurat [17].

Una din modalitățile de realizare a căutarii substructurilor active biologic este identificarea invarianților moleculari. În acest sens metoda WHIM (Weighted Holistic Invariant Molecular) calculează un set de indicatori statistici derivat din proprietățile sterice şi electrostatice ale moleculelor [28-30]. O variantă a acestei metode, MS-WHIM (de la Molecular Surface) serveşte în analiza suprafeței moleculare [31]. MS-WHIM este o colecție de 36 indicatori statistici derivați din proprietățile sterice şi electrostatice şi este orientată spre parametrizarea suprafeței moleculare [32].

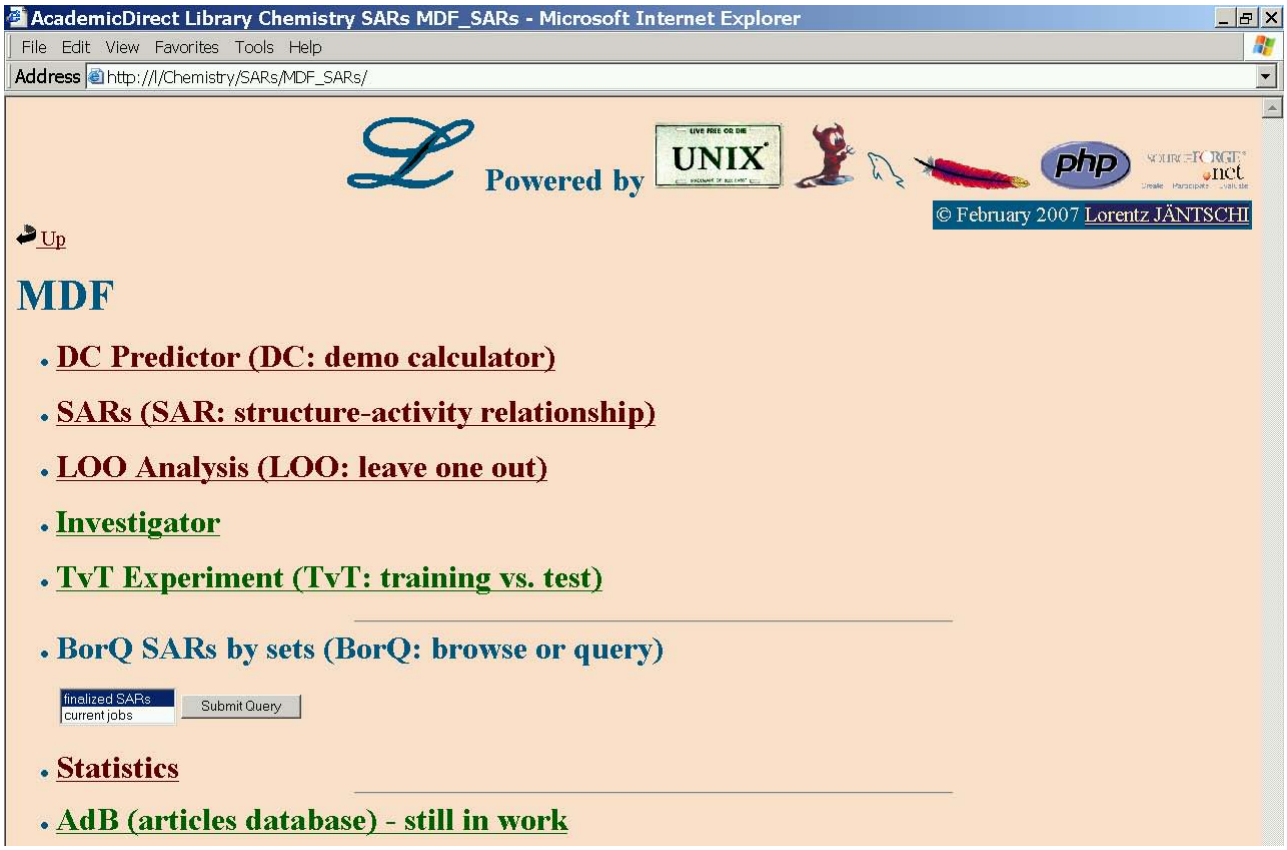
Relațiile cantitative structură-activitate QSAR (Q - quantitative) referă în general o activitate biologică măsurabilă iar tehnica de investigare ce foloseşte QSAR este o tehnică modernă folosită astăzi în multe domenii prioritare, incluzând farmaceutica, mediul, biotehnologia şi microbiologia. Literatura de specialitate conține astăzi un numar mare de relații structură-activitate care evaluează impactul produselor chimice asupra mediului [33-37].

Modelul matematic folosit pentru obținerea relațiilor structură-activiate este cel mai frecvent bazat pe regresia liniara [38] şi pe rețelele neuronale artificiale [39-41].

Rezultate obținute

Activitățile derulate în cadrul proiectului au permis atingerea obiectivelor propuse. Scopul cercetării a fost atins: *Implementarea unui sistem integrat online care să permită investigații structurale pe compusi biologic activi.*

Sistemul se află la adresa http://l.academicdirect.org, iar captura de ecran a acestuia este următoarea:



Aşa cum se poate observa şi din figura de mai sus, derularea propectului a permis şi identificarea viitoarele aplicații ale acestui sistem în integrarea cunoştințelor structurale pe compuşi biologic activi, şi anume construirea unei baze de date cu articole ştiințifice care tratează subiectul relațiilor structură-activitate pe compuşi biologic activi. O activiate preliminară transpunerii sistemului online a reprezentat-o achiziția, instalarea, testarea şi configurarea aparaturii suport. Finalitatea acestei activități a constituit-o serverul web care gazduieşte sistemul implementat. Caracteristicile acestui server sunt redate statistic în continuare:

193.226.7.203

CPU [100%]				
User	Nice	System	Interrupt	Idle
0	0	1	0	99

Idle

Mem [1111Mb]					
Active	Inact	Wired	Cache	Buf	Free
68	388	115	0	112	428

Inact	Free

OS:

FreeBSD 7.0-CURRENT
Copyright (c) 1992-2007 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994
The Regents of the University of California. All rights reserved.
FreeBSD is a registered trademark of The FreeBSD Foundation.
Up 11 days

kernel:

version:
FreeBSD 7.0-CURRENT #0
source:
root@l.academicdirect.ro:/usr/src/sys/i386/compile/LORI
borned:
Sat Feb 10 16:16:38 EET 2007

architecture:

i386

hostname:

l.academicdirect.ro

CPU:

CPU: Intel(R) Pentium(R) 4 CPU 2.80GHz (2790.71-MHz 686-class CPU)
cpu0: [ACPI CPU] on acpi0
p4tcc0: [CPU Frequency Thermal Control] on cpu0
Origin = "GenuineIntel" Id = 0xf27 Stepping = 7
Features = 0xbfebfbff [FPU, VME, DE, PSE, TSC, MSR, PAE, MCE, CX8, APIC, SEP, MTRR, PGE, MCA, CMOV, PAT, PSE36, CLFLUSH, DTS, ACPI, MMX, FXSR, SSE, SSE2, SS, HTT, TM, PBE]
Features2 = 0x4400 [CNXT-ID, XTPR]

Timecounters:

Timecounter "i8254" frequency 1193182 Hz quality 0
Timecounter "ACPI-fast" frequency 3579545 Hz quality 1000
Timecounter "TSC" frequency 2790712248 Hz quality 800
Timecounters tick every 1.000 msec

memory:

real memory = 1072627712 (1022 MB)
avail memory = 1044701184 (996 MB)

storage:

physical	device	size	free	mount at	type
drive	ad0	38166MB [Seagate ST340017A 3.31]		ata0-master	UDMA100
slice	ad0s2b	2048 Mb	2048 Mb	none	swap
slice	ad0s2a	29086 Mb	20520 Mb	/	file

devices:

name	link	device
vgapci0	at device 2.0 on pci0	[VGA-compatible display] mem 0x88000000-0x8fffffff, 0x80000000-0x8007ffff irq 11
pcib1	at device 30.0 on pci0	[ACPI PCI-PCI bridge]
pcib0	on acpi0	[ACPI Host-PCI bridge] port 0xcf8-0xcff
pci2	on pcib1	[ACPI PCI bus]
pci0	at device 29.7	[serial bus, USB]
pci0	at device 29.2	[serial bus, USB]
pci0	at device 29.1	[serial bus, USB]
pci0	at device 29.0	[serial bus, USB]
pci0	at device 31.3	[serial bus, SMBus]
pci0	at device 31.5	[multimedia, audio]
pci0	on pcib0	[ACPI PCI bus]
isab0	at device 31.0 on pci0	[PCI-ISA bridge]
fxp0	at device 8.0 on pci2	[Intel 82801DB (ICH4) Pro/100 VE Ethernet] port 0x2000-0x203f mem 0xc0100000-0xc0100fff irq 11
atapci0	at device 31.1 on pci0	[Intel ICH4 UDMA100 controller] port 0x1f0-0x1f7, 0x3f6, 0x170-0x177, 0x376, 0x1860-0x186f
ata1	on atapci0	[ATA channel 1]
ata0	on atapci0	[ATA channel 0]

network:

Interface	Network	Address	Ipkts	Ierrs	Opkts	Oerrs	Coll
fxp0	[Link#1]	00:09:6b:c6:14:99	5747009	0	1953541	0	0
fxp0	fe80:1::/64	fe80:1::209:6bff:	4	-	15562	-	-
fxp0	193.226.7.128	1.academicdirect.	2509282	-	1939978	-	-
fxp0	2001:b30:5000	2001:b30:5000:40:	0	-	6	-	-
lo0	-	[Link#2]	4975	0	4975	0	0
lo0	localhost	::1	1280	-	1280	-	-
lo0	fe80:2::/64	fe80:2::1	0	-	0	-	-
lo0	your-net	localhost	752	-	752	-	-

Pachetele soft care au fost instalate pentru a asigura suportul informatic necesar dezvoltării sistemului web online au fost:

- *apache-2.2.4* Version 2.2 of Apache web server with prefork MPM.
- *autoconf-2.59_2* Automatically configure source code on many Un*x platforms
- *bash-2.05b.007_6* The GNU Bourne Again Shell
- *cvsup-without-gui-16.1h_2* General network file distribution system optimized for CVS
- *expat-2.0.0_1* XML 1.0 parser written in C
- *fontconfig-2.3.2_6,1* An XML-based font configuration API for X Windows
- *freetype2-2.2.1_1* A free and portable TrueType font rendering engine
- *gawk-3.1.1_1* The GNU version of Awk
- *gettext-0.14.5_2* GNU gettext package
- *gmake-3.81_1* GNU version of 'make' utility
- *help2man-1.36.4_1* Automatically generating simple manual pages from program o
- *imake-6.9.0_1* Imake and other utilities from X.Org
- *jpeg-6b_4* IJG's jpeg compression utilities

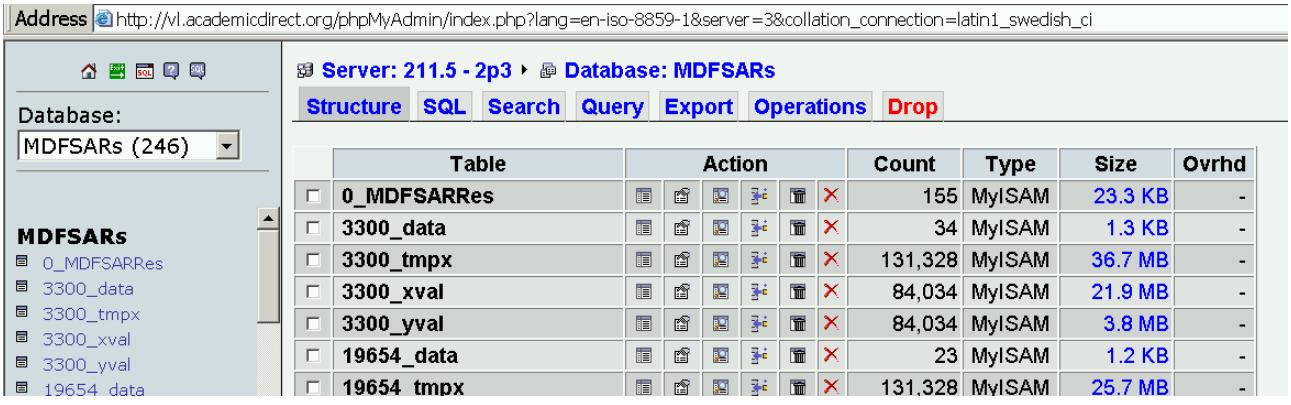
- *libdrm-2.0.2* Userspace interface to kernel Direct Rendering Module servi
- *libiconv-1.9.2_2* A character set conversion library
- *libtool-1.5.22_3* Generic shared library support script
- *libxml2-2.6.27* XML parser library for GNOME
- *m4-1.4.8_1* GNU m4
- *mc-light-4.1.40.p9_6* A lightweight Midnight Commander clone
- *memtest-4.0.6* Utility to test for faulty memory subsystem
- *mysql-client-5.1.14* Multithreaded SQL database (client)
- *mysql-server-3.23.59.n.20050301_3* Multithreaded SQL database (server)
- *p5-gettext-1.05_1* Message handling functions
- *perl-5.8.8* Practical Extraction and Report Language
- *php5-5.2.0* PHP Scripting Language (Apache Module and CLI)
- *php5-gd-5.2.0* The gd shared extension for php
- *php5-mysql-5.2.0* The mysql shared extension for php
- *pkg-config-0.21* A utility to retrieve information about installed libraries
- *png-1.2.14* Library for manipulating PNG images
- *t1lib-5.1.0_1,1* A Type 1 Rasterizer Library for UNIX/X11
- *xorg-libraries-6.9.0_1* X11 libraries and headers from X.Org

Pentru realizarea sistemului online propus a fost necesară contruirea de baze de date care să stocheze investigațiile structură-activitate efectuate.

S-a ales varianta stocării acestor informații pe un server din rețeaua locală a serverului 193.226.7.211 (172.27.211.1), şi anume pe calculatorul cu IP-ul 172.27.211.5.

Două baze de date au fost astfel construite. Una stochează seturile finalizate cu relații structură-activitate iar cealaltă seturile în lucru, urmand ca la finalizarea investigațiilor, aceste informații să se mute dintr-o bază de date în cealaltă.

La finalizarea proiectului ET36, baza de date cu seturi finalizate conține 246 de tabele, aşa cum se poate observa şi din figura de mai jos:



Statistica de umplere a acestor table este:

Nr	Tabel	Înregistrări	Mărime
1	0_MDFSARRes	155	23.3 KB
2	3300_data	34	1.3 KB
3	3300_tmpx	131,328	36.7 MB
4	3300_xval	84,034	21.9 MB
5	3300_yval	84,034	3.8 MB
6	19654_data	23	1.2 KB
7	19654_tmpx	131,328	25.7 MB
8	19654_xval	85,351	15.1 MB
9	19654_yval	85,351	3.9 MB
10	22583_data	57	1.5 KB
11	22583_tmpx	131,328	59.8 MB
12	22583_xval	82,105	35.8 MB
13	22583_yval	82,105	3.8 MB
14	23110_data	69	1.6 KB
15	23110_tmpx	131,328	71.8 MB
16	23110_xval	72,203	38.1 MB
17	23110_yval	72,203	3.3 MB
18	23158c_data	39	1.3 KB
19	23158c_tmpx	131,328	41.7 MB
20	23158c_xval	98,114	29.3 MB
21	23158c_yval	98,114	4.5 MB
22	23159e_data	8	1.1 KB
23	23159e_tmpx	131,328	10.7 MB
24	23159e_xval	99,816	6.2 MB
25	23159e_yval	99,816	4.6 MB
26	23167_data	31	1.3 KB
27	23167_tmpx	131,328	33.7 MB
28	23167_xval	95,123	22.6 MB
29	23167_yval	95,123	4.4 MB
30	26449t_data	10	1.1 KB
31	26449t_tmpx	131,328	12.7 MB
32	26449t_xval	94,843	7.3 MB

33	26449t_yval	94,843	4.3 MB
34	31572_data	24	1.2 KB
35	31572_tmpx	131,328	26.7 MB
36	31572_xval	111,477	20.5 MB
37	31572_yval	111,477	5.1 MB
38	33504_data	73	1.6 KB
39	33504_tmpx	131,328	75.8 MB
40	33504_xval	70,334	39.2 MB
41	33504_yval	70,334	3.2 MB
42	34121_data	76	1.7 KB
43	34121_tmpx	131,328	78.8 MB
44	34121_xval	90,190	52.4 MB
45	34121_yval	90,190	4.1 MB
46	34121a_data	76	1.7 KB
47	34121a_tmpx	131,328	78.8 MB
48	34121a_xval	90,181	52.4 MB
49	34121a_yval	90,181	4.1 MB
50	34121bad_data	76	1.7 KB
51	34121bad_tmpx	131,328	78.8 MB
52	34121bad_xval	90,267	52.4 MB
53	34121bad_yval	90,267	4.1 MB
54	34121nopt_data	76	1.7 KB
55	34121nopt_tmpx	131,328	78.8 MB
56	34121nopt_xval	90,195	52.4 MB
57	34121nopt_yval	90,195	4.1 MB
58	36638_data	16	1.1 KB
59	36638_tmpx	131,328	18.7 MB
60	36638_xval	105,319	13.0 MB
61	36638_yval	105,319	4.8 MB
62	41521_data	8	1.1 KB
63	41521_tmpx	131,328	10.7 MB
64	41521_xval	86,407	5.4 MB
65	41521_yval	86,407	4.0 MB
66	52344_data	8	1.1 KB

67	52344_tmpx	131,328	10.7 MB
68	52344_xval	96,769	6.0 MB
69	52344_yval	96,769	4.4 MB
70	52730_data	10	1.1 KB
71	52730_tmpx	131,328	12.7 MB
72	52730_xval	93,090	7.2 MB
73	52730_yval	93,090	4.3 MB
74	408461_data	40	1.4 KB
75	408461_tmpx	131,328	42.7 MB
76	408461_xval	70,943	21.7 MB
77	408461_yval	70,943	3.2 MB
78	408462_data	40	1.4 KB
79	408462_tmpx	131,328	42.7 MB
80	408462_xval	70,954	21.7 MB
81	408462_yval	70,954	3.2 MB
82	408464_data	40	1.4 KB
83	408464_tmpx	131,328	42.7 MB
84	408464_xval	70,943	21.7 MB
85	408464_yval	70,943	3.2 MB
86	DHFR_data	67	1.6 KB
87	DHFR_tmpx	131,328	69.8 MB
88	DHFR_xval	89,164	45.7 MB
89	DHFR_yval	89,164	4.1 MB
90	DevMTO_tmpx	131,328	10.7 MB
91	DevMTOp00_data	8	1.1 KB
92	DevMTOp00_tmpx	131,328	10.7 MB
93	DevMTOp00_xval	72,288	4.5 MB
94	DevMTOp00_yval	72,288	3.3 MB
95	DevMTOp01_data	5	1.0 KB
96	DevMTOp01_tmpx	131,328	7.7 MB
97	DevMTOp01_xval	71,682	2.8 MB
98	DevMTOp01_yval	71,682	3.3 MB
99	DevMTOp02_data	7	1.1 KB
100	DevMTOp02_tmpx	131,328	9.7 MB

101	DevMTOp02_xval	71,922	3.9 MB
102	DevMTOp02_yval	71,922	3.3 MB
103	DevMTOp03_data	8	1.1 KB
104	DevMTOp03_tmpx	131,328	10.7 MB
105	DevMTOp03_xval	72,208	4.5 MB
106	DevMTOp03_yval	72,208	3.3 MB
107	DevMTOp04_data	8	1.1 KB
108	DevMTOp04_tmpx	131,328	10.7 MB
109	DevMTOp04_xval	72,220	4.5 MB
110	DevMTOp04_yval	72,220	3.3 MB
111	DevMTOp05_data	8	1.1 KB
112	DevMTOp05_tmpx	131,328	10.7 MB
113	DevMTOp05_xval	72,215	4.5 MB
114	DevMTOp05_yval	72,215	3.3 MB
115	DevMTOp06_data	7	1.1 KB
116	DevMTOp06_tmpx	131,328	9.7 MB
117	DevMTOp06_xval	71,908	3.9 MB
118	DevMTOp06_yval	71,908	3.3 MB
119	DevMTOp07_data	8	1.1 KB
120	DevMTOp07_tmpx	131,328	10.7 MB
121	DevMTOp07_xval	72,280	4.5 MB
122	DevMTOp07_yval	72,280	3.3 MB
123	DevMTOp08_data	7	1.1 KB
124	DevMTOp08_tmpx	131,328	9.7 MB
125	DevMTOp08_xval	71,888	3.9 MB
126	DevMTOp08_yval	71,888	3.3 MB
127	DevMTOp09_data	7	1.1 KB
128	DevMTOp09_tmpx	131,328	9.7 MB
129	DevMTOp09_xval	71,902	3.9 MB
130	DevMTOp09_yval	71,902	3.3 MB
131	DevMTOp10_data	7	1.1 KB
132	DevMTOp10_tmpx	131,328	9.7 MB
133	DevMTOp10_xval	73,908	4.0 MB
134	DevMTOp10_yval	73,908	3.4 MB

135	DevMTOp11_data	6	1.1 KB
136	DevMTOp11_tmpx	131,328	8.7 MB
137	DevMTOp11_xval	71,909	3.4 MB
138	DevMTOp11_yval	71,909	3.3 MB
139	DevMTOp12_data	8	1.1 KB
140	DevMTOp12_tmpx	131,328	10.7 MB
141	DevMTOp12_xval	72,253	4.5 MB
142	DevMTOp12_yval	72,253	3.3 MB
143	DevMTOp14_data	8	1.1 KB
144	DevMTOp14_tmpx	131,328	10.7 MB
145	DevMTOp14_xval	72,287	4.5 MB
146	DevMTOp14_yval	72,287	3.3 MB
147	DevMTOp15_data	6	1.1 KB
148	DevMTOp15_tmpx	131,328	8.7 MB
149	DevMTOp15_xval	71,935	3.4 MB
150	DevMTOp15_yval	71,935	3.3 MB
151	DevMTOp16_data	8	1.1 KB
152	DevMTOp16_tmpx	131,328	10.7 MB
153	DevMTOp16_xval	72,281	4.5 MB
154	DevMTOp16_yval	72,281	3.3 MB
155	DevMTOp17_data	8	1.1 KB
156	DevMTOp17_tmpx	131,328	10.7 MB
157	DevMTOp17_xval	72,285	4.5 MB
158	DevMTOp17_yval	72,285	3.3 MB
159	DevMTOp18_data	6	1.1 KB
160	DevMTOp18_tmpx	131,328	8.7 MB
161	DevMTOp18_xval	71,890	3.4 MB
162	DevMTOp18_yval	71,890	3.3 MB
163	DevMTOp19_data	7	1.1 KB
164	DevMTOp19_tmpx	131,328	9.7 MB
165	DevMTOp19_xval	71,916	3.9 MB
166	DevMTOp19_yval	71,916	3.3 MB
167	DevMTOp20_data	8	1.1 KB
168	DevMTOp20_tmpx	131,328	10.7 MB

169	DevMTOp20_xval	72,252	4.5 MB
170	DevMTOp20_yval	72,252	3.3 MB
171	DevMTOp21_data	8	1.1 KB
172	DevMTOp21_tmpx	131,328	10.7 MB
173	DevMTOp21_xval	72,226	4.5 MB
174	DevMTOp21_yval	72,226	3.3 MB
175	DevMTOp22_data	7	1.1 KB
176	DevMTOp22_tmpx	131,328	9.7 MB
177	DevMTOp22_xval	71,906	3.9 MB
178	DevMTOp22_yval	71,906	3.3 MB
179	DevMTOp23_data	8	1.1 KB
180	DevMTOp23_tmpx	131,328	10.7 MB
181	DevMTOp23_xval	72,290	4.5 MB
182	DevMTOp23_yval	72,290	3.3 MB
183	DevMTOp24_data	7	1.1 KB
184	DevMTOp24_tmpx	131,328	9.7 MB
185	DevMTOp24_xval	71,894	3.9 MB
186	DevMTOp24_yval	71,894	3.3 MB
187	DevMTOp25_data	7	1.1 KB
188	DevMTOp25_tmpx	131,328	9.7 MB
189	DevMTOp25_xval	71,901	3.9 MB
190	DevMTOp25_yval	71,901	3.3 MB
191	Dipeptides_data	58	1.5 KB
192	Dipeptides_tmpx	131,328	60.8 MB
193	Dipeptides_xval	93,310	41.4 MB
194	Dipeptides_yval	93,310	4.3 MB
195	IChr10_data	10	1.1 KB
196	IChr10_tmpx	131,328	12.7 MB
197	IChr10_xval	103,237	8.0 MB
198	IChr10_yval	103,237	4.7 MB
199	JCCS2001_data	47	1.4 KB
200	JCCS2001_tmpx	131,328	49.8 MB
201	JCCS2001_xval	88,957	32.4 MB
202	JCCS2001_yval	88,957	4.1 MB

203	MR10_data	10	1.1 KB
204	MR10_tmpx	131,328	12.7 MB
205	MR10_xval	107,692	8.3 MB
206	MR10_yval	107,692	4.9 MB
207	PCB_lkow_data	206	2.8 KB
208	PCB_lkow_tmpx	131,328	209.1 MB
209	PCB_lkow_xval	62,750	98.7 MB
210	PCB_lkow_yval	62,750	2.9 MB
211	PCB_rrf_data	209	2.8 KB
212	PCB_rrf_tmpx	131,328	212.1 MB
213	PCB_rrf_xval	62,873	100.3 MB
214	PCB_rrf_yval	62,873	2.9 MB
215	PCB_rrt_data	209	2.8 KB
216	PCB_rrt_tmpx	131,328	212.1 MB
217	PCB_rrt_xval	62,712	100.1 MB
218	PCB_rrt_yval	62,712	2.9 MB
219	RRC433_lbr_data	30	1.3 KB
220	RRC433_lbr_tmpx	131,328	32.7 MB
221	RRC433_lbr_xval	86,409	19.9 MB
222	RRC433_lbr_yval	86,409	4.0 MB
223	RRC433_lkow_data	30	1.3 KB
224	RRC433_lkow_tmpx	131,328	32.7 MB
225	RRC433_lkow_xval	86,388	19.9 MB
226	RRC433_lkow_yval	86,388	4.0 MB
227	RRC433_pka_data	30	1.3 KB
228	RRC433_pka_tmpx	131,328	32.7 MB
229	RRC433_pka_xval	86,373	19.9 MB
230	RRC433_pka_yval	86,373	4.0 MB
231	Ta395_data	15	1.1 KB
232	Ta395_tmpx	131,328	17.7 MB
233	Ta395_xval	102,608	11.8 MB
234	Ta395_yval	102,608	4.7 MB
235	Tox395_data	14	1.1 KB
236	Tox395_tmpx	131,328	16.7 MB

237	Tox395_xval	103,411	11.1 MB
238	Tox395_yval	103,411	4.7 MB
239	Triazines_data	30	1.3 KB
240	Triazines_tmpx	131,328	32.7 MB
241	Triazines_xval	74,467	17.1 MB
242	Triazines_yval	74,467	3.4 MB
243	a_acids_data	12	1.1 KB
244	a_acids_tmpx	131,328	14.7 MB
245	a_acids_xval	97,007	9.0 MB
246	a_acids_yval	97,007	4.4 MB
Total	246 tabele	18,063,808	3.5 GB

Baza de date cu seturi în lucru (`MDFSARtmp`) conține 61 de tabele:

Nr	Tabel	Înregistrări	Mărime
1	0_MDFSARRes	25	6.6 KB
2	15aacidsCHI_data	15	1.1 KB
3	15aacidsCHI_tmpx	131,328	17.7 MB
4	15aacidsCHI_xval	101,396	11.8 MB
5	15aacidsCHI_yval	101,398	4.6 MB
6	15aacidsDM_data	15	1.1 KB
7	15aacidsDM_tmpx	131,328	17.7 MB
8	15aacidsDM_xval	101,428	11.8 MB
9	15aacidsDM_yval	101,428	4.6 MB
10	15aacidsEHu_data	15	1.1 KB
11	15aacidsEHu_tmpx	131,328	17.7 MB
12	15aacidsEHu_xval	101,450	11.8 MB
13	15aacidsEHu_yval	101,450	4.6 MB
14	15aacidsHTH_data	15	1.1 KB
15	15aacidsHTH_tmpx	131,328	17.7 MB
16	15aacidsHTH_xval	101,442	11.8 MB
17	15aacidsHTH_yval	101,442	4.6 MB
18	15aacidsHyE_data	15	1.1 KB
19	15aacidsHyE_tmpx	131,328	17.7 MB
20	15aacidsHyE_xval	101,396	11.8 MB

21	15aacidsHyE_yval	101,396	4.6 MB
22	15aacidsHyd_data	15	1.1 KB
23	15aacidsHyd_old_xval	101,306	13.6 MB
24	15aacidsHyd_old_yval	101,306	6.6 MB
25	15aacidsHyd_tmpx	131,328	17.7 MB
26	15aacidsHyd_xval	101,306	11.8 MB
27	15aacidsHyd_yval	101,306	4.6 MB
28	15aacidsKDH_data	15	1.1 KB
29	15aacidsKDH_tmpx	131,328	17.7 MB
30	15aacidsKDH_xval	101,417	11.8 MB
31	15aacidsKDH_yval	101,417	4.6 MB
32	15aacidsLPH_data	15	1.1 KB
33	15aacidsLPH_tmpx	131,328	17.7 MB
34	15aacidsLPH_xval	101,430	11.8 MB
35	15aacidsLPH_yval	101,430	4.6 MB
36	15aacidsLac_data	15	1.1 KB
37	15aacidsLac_tmpx	131,328	17.7 MB
38	15aacidsLac_xval	101,431	11.8 MB
39	15aacidsLac_yval	101,431	4.6 MB
40	15aacidsLogP_data	15	1.1 KB
41	15aacidsLogP_tmpx	131,328	17.7 MB
42	15aacidsLogP_xval	101,416	11.8 MB
43	15aacidsLogP_yval	101,416	4.6 MB
44	15aacidsMR_data	15	1.1 KB
45	15aacidsMR_tmpx	131,328	17.7 MB
46	15aacidsMR_xval	101,457	11.8 MB
47	15aacidsMR_yval	101,464	4.6 MB
48	15aacidsPol_data	15	1.1 KB
49	15aacidsPol_tmpx	131,328	17.7 MB
50	15aacidsPol_xval	101,452	11.8 MB
51	15aacidsPol_yval	101,452	4.6 MB
52	15aacidsRef_data	15	1.1 KB
53	15aacidsRef_tmpx	131,328	17.7 MB
54	15aacidsRef_xval	101,458	11.8 MB

55	15aacidsRef_yval	101,475	4.6 MB
56	15aacidsSlb_data	15	1.1 KB
57	15aacidsSlb_tmpx	131,328	17.7 MB
58	15aacidsSlb_xval	101,427	11.8 MB
59	15aacidsSlb_yval	101,427	4.6 MB
60	15aacids_tmpx	131,328	17.7 MB
61	ready	14	1.4 KB
62	61 tabele	5,012,619	516.0 MB

Relațiile structură-activitate stocate în baza de date cu activități biologice finalizate este redată în tabelul de mai jos:

Nr	Set activitate	Ecuație MDF SAR	Coeficient de determinare	nr. Var.	nr. Mol.
1	IChr10_	y=-62.361371746820275 + lSDmwMt*6.372504386827629 + iHPDEQg*0.058693170881799	0.999221607	2	10
2	IChr10_	y=20.460137586773501 + lHMrtCt*-6.961293758269948 + iBPmTEt*-969.172867908018304	0.998805743	2	10
3	36638_	y=2.582556947454996 + lPMDVQg*0.002969357191374 + IsPrVHg*-22.592449746207965	0.977605386	2	16
4	36638_	y=2.585556625918434 + lPMDVQg*0.002984666705928 + IsPmVHg*-22.108669187753978	0.976781931	2	16
5	36638_	y=2.577163501835581 + IiMMWHt*0.853020387909027 + lPMDVQg*0.002947685825620	0.981091504	2	16
6	36638_	y=2.745069935843683 + IiMMWHt*0.964882719278833 + IFMmkHg*0.002015868714906 + LPDMVQg*0.002704618929348	0.992472044	3	16
7	36638_	y=2.568663834381964 + IiMMWHt*0.857321758795370 +	0.994462046	3	16

		lPMDVQg*0.002952308975991 + iFMdFQg*0.000000000000802			
8	36638_	y=2.572960050395093 + IiMMWHt*0.862026691653915 + lPMDVQg*0.002975951926473 + iSPMtQg*0.000359275274073	0.994961682	3	16
9	36638_	y=2.573205449280688 + IiMMWHt*0.862522334633554 + lPMDVQg*0.002984079061001 + ibPMtQg*0.000060106252532	0.995018995	3	16
10	PCB_rrf_	y=5.085278949837590 + imMrFHt*-357.296321414089600 + iHDdFHg*2.156138113196185	0.692921683	2	209
11	PCB_rrf_	y=6.055340736461524 + imMrFHt*-416.942003738513088 + iHDdFHg*2.313956389789302 + iMMMjQg*1.829475297508006 + iAMrVQg*-0.002506945942322	0.736793462	4	209
12	23159e_	y=16.624223709106445 + lfDMWHt*-0.216979935765266 + IbmrTEt*-0.683317601680756	0.98164165	2	14
13	23159e_	y=18.090740203857424 + lfDMWHt*-0.217230200767517 + IbmrtEt*-0.770600378513336	0.981111884	2	14
14	Ta395_	y=-4.488118053335837 + INDRLQt*8.348191853294373 + lHPmTMt*1.965121565280751	0.976574132	2	15
15	Tox395_	y=-1.293780346589664 + liMrSQg*0.223497432069395 + ASPrVQg*0.095534977438483	0.95585816	2	14
16	Tox395_	y=-1.272088763676020 + lIMrSQg*0.223829242224039 + ASPrVQg*0.095600357852137	0.955056854	2	14
17	Tox395_	y=-0.717673381133009 +	0.957735266	2	14

		lsMrSQg*0.225050651084089 + ASPrVQg*0.098706316363904			
18	Tox395_	y=-1.575004492205578 + lNMrSQg*0.205885242323993 + ASPrVQg*0.092984460101768	0.956820302	2	14
19	Tox395_	y=-1.597971465629698 + lNMrEQg*0.336843860556055 + ASPrVQg*0.094660231172752	0.955928148	2	14
20	41521_	y=15.977082257070075 + lAPRkHg*4.411577306950580 + iAPmEQg*-0.004799303305105	0.998675191	2	8
21	41521_	y=15.557416220778391 + lHPDKHg*-4.447021764795913 + iBPmEQg*-0.045008303244039	0.9990247	2	8
22	41521_	y=15.977082257072571 + lHPRKHg*-4.411577306951695 + iAPmEQg*-0.004799303305106	0.998675191	2	8
23	41521_	y=43.344306644672435 + ImMdsEg*-2.208255961271429 + lIMMFQt*3.740518408044116	0.999125495	2	8
24	MR10_	y=19.406881564985923 + lGDrtMg*21.873029005600086 + lAmrVGg*-164.023794938214048	0.999797892	2	10
25	MR10_	y=17.394528468178550 + lGDmSMt*28.247433444091738 + lAmrfEt*-83.965315146143757	0.999958406	2	10
26	26449t_	y=-19.112864129011363 + lHPDOQg*2.317497007317971 + IsMRKGg*19.343096061022083	0.997345142	2	10
27	52344_	y=-5.342090912944618 + ISDrSQg*-0.000836073645555 + iSmrJQt*-33.145388286106125	0.999457098	2	8
28	52344_	y=-13.259851363534467 + IAPdwCg*7.421342259908507 + lSDMkMg*-	0.99978832	2	8

		1.203325361332910			
29	52344_	y=7.182585269595931 + lbPMkHg*-1.097130469643120 + iAPrVGt*-33.244348016071162	0.999803127	2	8
30	DevMTOp00_	y=82487.779927956544000 + iSDmtQg*-4291.591445441648640 + lAMrFEt*-24751.302374423369600	0.999997516	2	8
31	DevMTOp01_	y=-5.211074704460113 + IAPMLMt*0.000744091093604 + lmPmlQg*2.674299018254657	1	2	5
32	DevMTOp02_	y=8.572782794598863 + anPrdQg*-0.062403551223709 + IsmRLQt*233.992406665280768	0.999966002	2	7
33	DevMTOp03_	y=21.577675933246624 + iHDRkMg*-0.004324723908977 + inMrPQg*11.108974534804655	0.999152012	2	8
34	DevMTOp04_	y=6.946100250910472 + iGDREHg*-10.090383646214632 + lnDDVQg*-1.391295355575833	0.999779572	2	8
35	DevMTOp05_	y=5.972901448850221 + AHDmtQg*382.966792379103488 + inMDqQg*-5.152599794558495	0.9992016	2	8
36	DevMTOp06_	y=4.258542471386298 + lGPMqGg*-55.309513051550080 + IBPrJHt*-0.016924218996259	0.999981768	2	7
37	DevMTOp07_	y=16.997960609979277 + iAMrECt*-14.724870628170749 + aAPmfQt*-0.105749387186883	0.999603803	2	8
38	DevMTOp08_	y=2.299452498508210 + LsDmjQg*-1.172330969315326 + lAMrDHt*-6.960913760516673	0.999959098	2	7
39	DevMTOp09_	y=49.601873825853658 + IAMrDQg*-0.353928603916190 + iBPRwMt*31653.215583361798400	0.999903215	2	7

40	DevMTOp10_	y=-4.563145600075828 + isMdsGg*-0.076556062852060 + lsDDkQg*-1.431507685251379	0.999898517	2	7
41	DevMTOp11_	y=-39.637026030878438 + iGPMpEg*14.697649143241879 + lPDDpEg*0.044063759116077	0.9999946	2	6
42	DevMTOp12_	y=-252.938700802085632 + lMMRSGt*24.401713743828797 + lsPmpMg*-28.277181303483437	0.999253357	2	8
43	DevMTOp14_	y=28.031130403667130 + IHMRFEg*-0.897288952138859 + INPmsQt*-13455.651827727787520	0.99957823	2	8
44	DevMTOp15_	y=-269.701345715687040 + iIMMoEg*32.790755969495002 + IHmrlEt*1058.588149821219040	0.999995588	2	6
45	DevMTOp16_	y=-4.591222054726206 + asDmkQg*0.003446474307407 + IGMmTHt*2.246662887026612	0.99894463	2	8
46	DevMTOp17_	y=0.276515007162331 + iIPdqQg*1.456930890306632 + iImrSCg*-0.008375218150532	0.999544921	2	8
47	DevMTOp18_	y=5.277557317326950 + iGMmSQt*-0.925518492017054 + IIPdwQg*-561.357019867305216	0.999999653	2	6
48	DevMTOp19_	y=484.609231791737472 + iADDOMg*-6893.174261695466240 + lmPmLMt*-52.452348531199731	0.999942872	2	7
49	DevMTOp20_	y=-76.465419087381811 + lGPrfGt*11.765097022456481 + iGPMqMg*14.548142362264151	0.999179456	2	8
50	DevMTOp21_	y=-0.466773345458314 + ISPRfEt*0.658065324549722 + imDrwEt*-688.623604086868352	0.999019748	2	8
51	DevMTOp22_	y=35.886964607142445 +	0.999927658	2	7

		lIDrdQg*8.293607635627836 + isMdPQg*0.001557300106484			
52	DevMTOp23_	y=-267.218704057874592 + lsPmkEg*-20.968463155992515 + lSmRFGt*51.184666540653107	0.999864996	2	8
53	DevMTOp24_	y=0.355538934447154 + IsPDPHg*99.897075577592973 + InDRLQg*-0.736739109598832	0.9999568	2	7
54	DevMTOp25_	y=1.893045064859439 + lBMDVQg*4.089050756254091 + iHDrDQt*-42.497588982800794	0.999931515	2	7
55	RRC433_lbr_	y=-2.260942771393344 + ASMmVQt*0.036537920618985 + lfDdOQg*-0.216319157172228	0.897264116	2	30
56	RRC433_lbr_	y=-3.294716228841605 + ASMmVQt*0.034759339883385 + lfDdOQg*-0.326175236962887 + InMrLQg*0.079023124090750 + LsDMpQg*-0.346227342563766	0.973702913	4	30
57	RRC433_lbr_	y=-3.309083909227967 + ASMmVQt*0.035359451252071 + lfDdOQg*-0.326340537986437 + InMrLQt*0.083521027478412 + LsDMpQg*-0.353965995278507	0.973908851	4	30
58	RRC433_lkow_	y=1.068504837149195 + isDDkGg*0.003385972696729 + IMmrKQg*-0.401342461954502	0.894339198	2	30
59	RRC433_lkow_	y=0.086887657603207 + isDDkGg*0.005559930394138 + IMmrKQg*-0.416458920866538 + lPMDKQg*0.009409040694936 + lFMMKQg*-0.077972151747497	0.978050343	4	30
60	RRC433_pka_	y=12.145778693743030 + AHMMVQg*-1.759312774646531 + inDmwHg*-	0.85106015	2	30

		1.423171133335602			
61	RRC433_pka_	y=12.250129572666588 + AHMMVQg*-1.878604996401587 + inDmwHg*-1.409105150926869 + IimRJQg*-1.255850637865408 + AsPrwQg*-551.820132942366464	0.96048211	4	30
62	RRC433_pka_	y=12.266395089058967 + AHMMVQg*-1.886964796625174 + inDmwHg*-1.431793246534204 + IImRJQg*-1.492725761439522 + IHDrsQg*-132.763291817560480	0.961458919	4	30
63	RRC433_pka_	y=13.187486747004316 + AHMMVQg*-1.628307288455412 + inDmwHg*-1.315727674279415 + LBMRlQg*-0.450392531615080 + AHDmEQg*-3.537408893070444	0.963779329	4	30
64	52730_	y=2.799163315535306 + IbMmpMg*28.064230202549747 + LPPROQg*0.081524293438233	0.99763089	2	10
65	DHFR_	y=3.982186750193776 + iImrKHt*1.516060326175799 + liMDWHg*2.357039692152132	0.85627596	2	67
66	DHFR_	y=-1.172577695940167 + lImrKHt*-4.863206399509672 + IiMDWHg*5.474142782322419	0.857180673	2	67
67	DHFR_	y=3.261481723647712 + lImrKHt*-4.900679297589359 + lIMDWHg*2.313684191877673	0.858873758	2	67
68	DHFR_	y=22.783498446732483 + IbMmSHt*-0.044781937978148 + inPRjHg*0.036908423922842 + lsMMTGg*2.276594353795319	0.887769256	3	67
69	DHFR_	y=3.766172816092689 + iImrKHt*1.630691978329184 +	0.905161273	4	67

		liMDWHg*2.398496510115706 + LSPmEQg*-0.085551239269939 + IIDrJQt*0.293720167343832			
70	DHFR_	y=3.781999644467336 + iImrKHt*1.619894894103214 + liMDWHg*2.371371074452369 + IsDrJQt*6.402293814267814 + LSPmEQg*-0.085177985460006	0.905784714	4	67
71	a_acids_	y=-5.746809646380628 + IiPdJHg*11.956269601549317 + iHMmVQg*0.065716945111961	0.984823373	2	12
72	a_acids_	y=11.256459289110492 + iHmrsGg*0.389578813720558 + iiPDOHg*-2.366249952762924	0.987102214	2	12
73	a_acids_	y=9.387440286015163 + lIPrVGt*6.088203345034852 + IHDDKHg*0.381482245798468	0.98835913	2	12
74	34121bad_	y=-8.951974935678216 + lIMdFGg*-9.901521373121243 + IAPmTCt*36.243596058596800	0.509261534	2	76
75	34121nopt_	y=13.360278625318943 + iIPRLGg*-4.412916051903299 + LiMmwQg*0.421902462393154	0.622248309	2	76
76	19654_	y=0.183306719129093 + IFDDpGg*-0.000068304771741 + ISDrFMt*-0.000001260853126	0.984019252	2	23
77	19654_	y=-0.008829461052500 + isDRtHg*-0.000051338532157 + iHMMtHg*0.130030477933933	0.988351291	2	23
78	19654_	y=0.046790454181503 + ismRSEg*-0.000172348201588 + isDRTCg*0.000030999802774	0.98955647	2	23
79	19654_	y=7.616547989246049 + lIDrpMg*1.001309641822330 + lsPrJGt*-	0.996248249	4	23

		1.069779866848350 + ismRSEg*-0.000163062657210 + isDRTCg*0.000029929545873			
80	19654_	y=2.733124886439780 + lmDDKMg*-0.286110586112610 + imPrdQt*-1.789020475582959 + ismRSEg*-0.000161361248412 + isDRTCg*0.000029725595809	0.997851065	4	23
81	19654_	y=6.884466558075200 + liMRsCg*0.921784722969737 + lIDRFMg*-1.317553545161115 + isDRtHg*-0.000050399261806 + iHMMtHg*0.135432046071406	0.996819148	4	23
82	19654_	y=-0.146575674440346 + imMRkMg*0.000902003747302 + imMDVQg*-0.322525717918453 + isDRtHg*-0.000052364343767 + iHMMtHg*0.139074684280391	0.997260489	4	23
83	3300_	y=17.314335672067008 + isMdTHg*0.002390228128569 + lbDrqHg*-12.425599812782971	0.917199024	2	34
84	3300_	y=-17.709380801328218 + isMdTHg*0.002306275036298 + IiDrQHg*77.216162936821005	0.918384193	2	34
85	3300_	y=-20.627769593100518 + IsDDQHg*-620.969281908775936 + AsMmjQg*3.240577436194556 + isMdTHg*0.002240837541751 + IiDrQHg*120.206085120454093	0.965508385	4	34
86	3300_	y=-17.092683014823635 + iHDmkQt*0.688855852114620 + AsDmtQg*205224.367519787520000 + isMdTHg*0.002310828590955 + IiDrQHg*72.349612191442458	0.966514933	4	34

87	22583_	y=-4.303662300109863 + lIDrFEg*-19.435102462768554 + IiMMsGg*11.070854187011718	0.782257676	2	57
88	22583_	y=-4.358335018157959 + lIDrFEg*-19.590654373168944 + IiMMSGg*11.005515098571777	0.783080041	2	57
89	22583_	y=16.697563171386717 + InMdTHg*-9.215051651000977 + lfDMwEt*-0.860860228538513 + AsMrKQt*141.058273315429680	0.835162103	3	57
90	22583_	y=-5.146468639373779 + lIDrFEg*-18.882848739624022 + IiMMSGg*9.991197586059570 + iFDmkHg*-0.000023631801014 + inMrEQt*-1.860737323760986	0.900917828	4	57
91	22583_	y=17.723573684692381 + InMdTHg*-7.113586425781251 + lFDMwEt*-1.234191298484802 + AiMrKQt*8.357679367065430 + ImDMtQt*659184.625000000000000 + lIMdEMg*-5.981501102447510	0.917499542	5	57
92	22583_	y=10.439790725708008 + InMdTHg*-6.621389389038086 + lFDMwEt*-1.115040302276611 + AiMrKQt*7.575278282165527 + IMDMtQt*782467.875000000000000 + iIMdTMg*18.747991561889648	0.917872667	5	57
93	33504_	y=-323.024772597417024 + liDmEHt*-105.929211454059789 + IADmwHt*17.760000133301725	0.998191949	2	73
94	33504_	y=-129.202920028528064 + lGDrtGt*-67.450781570303091 + IbDrfHt*4.889627699125125	0.998194885	2	73
95	23158c_	y=6.272650515627022 + iiMdLGg*-	0.949331559	2	40

		24.465622121014768 + AsPmVQt*7.416438629162940			
96	23158c_	y=-8.925568204084518 + lIMdLGg*6.533652861276372 + AsPmVQt*7.390652840339525	0.949405043	2	40
97	23158c_	y=-9.030826989310386 + lIMdLGg*6.602715759615876 + AIPmVQt*0.738833729895370	0.949450328	2	40
98	23158c_	y=-9.028650724880719 + lIMdLGg*6.601183808052042 + AiPmVQt*0.738839898477699	0.949451061	2	40
99	23158c_	y=7.413062019927823 + iIMdLGg*-29.203947244192070 + AsPmVQt*7.376749884182984	0.950945432	2	40
100	23158c_	y=7.483477154220049 + iIMdLGg*-29.525257834273594 + AIPmVQt*0.737465221651949	0.951085814	2	40
101	23158c_	y=7.481885945863868 + iIMdLGg*-29.518679478761894 + AiPmVQt*0.737471789565067	0.951086994	2	40
102	23158c_	y=6.372537492476300 + IBMrkGg*-92.371378276225190 + IsPmVQt*-7.281656762147562	0.955679462	2	40
104	31572_	y = 3.5140 + iNMRJQt*3.9523e-2 + iSDRkQt*-9.1485 + LsPrDQt*5.7624e-1 + IADRSHg*-1.1986e-1	0.958313838	4	24
105	31572_	y = -10.573 + IIMmSCg*63.140 + lIMRVCg*-5.1308	0.830588637	2	24
106	31572_	y = 3.9864 + LsPrDQt*6.4877e-1 + IADRSHg*-1.2942e-1	0.811092657	2	24
107	23110_	y=3.309258874208278 + ABmrtQg*-14.191657164410053 + iGPrfHt*0.960433156574692	0.870389569	2	69
108	23110_	y=3.255577576284918 + ABmrsQg*-	0.870762905	2	69

		9.655480534462673 + iGPrfHt*1.003843270199119			
109	23110_	y=3.468032235802128 + ABmrtQg*-14.109591667547505 + iGPrfHt*1.000897643171127 + iImrfQt*-0.000035842320734 + aHDRdQt*-0.019679357321205	0.923419436	4	69
110	23110_	y=3.649903008444890 + IBDMWQt*0.668182102284062 + iIPmWHt*0.139974134207539 + IMPrkQg*0.000238539437306 + aHPMwQt*-0.000458937750242	0.922112869	4	69
111	23110_	y=3.635644435882568 + ABDmtQg*-11.895344734191895 + ISDRLQt*0.000576242397074 + iIPrwHt*0.130846872925758	0.901068389	3	69
112	23110_	y=3.463929176330566 + ABDmtQg*-13.019453048706056 + iHPrwHt*1.618527054786682 + aIPMwQt*-0.000028133128581	0.902104378	3	69
113	23110_	y=3.494592428207398 + ABDmtQg*-13.096261978149414 + iHPmwHt*1.626412868499756 + aIPMwQt*-0.000028470527468	0.902782798	3	69
114	23110_	y=3.821168899536133 + aHPMwQt*-0.000338123209076 + iIPmWHt*0.132829502224922 + IBPMWQt*0.756949245929718	0.904430389	3	69
115	23110_	y=3.719008207321167 + aHPMwQt*-0.000501490896568 + iIPmWHt*0.142897620797157 + IBPMWQt*0.778551995754242 + iFMMfGt*-0.000000000418976 + IMPrkQg*0.000253266363870	0.936299264	5	69

116	23110_	y=3.719301223754883 + aHPMwQt*-0.000501317554154 + iIPmWHt*0.142888113856316 + IBPMWQt*0.778867542743683 + iFmRFMt*-0.000000309545300 + IMPrkQg*0.000253249396337	0.936396182	5	69
117	23167_	y=3.937699079513550 + iSPRtQg*0.051902804523706 + imDrkQt*-5.906542778015137	0.861876726	2	31
118	23167_	y=3.934797048568726 + aSPRtQg*0.051960568875074 + imDrkQt*-5.895822048187256	0.860438466	2	31
119	23167_	y=3.937698841094971 + imDrkQt*-5.906543731689453 + iSPRtQg*0.051902808248997	0.861882389	2	31
120	23167_	y=3.934797048568726 + imDrkQt*-5.895823001861572 + aSPRtQg*0.051960572600365	0.860438466	2	31
121	23167_	y=3.832609653472900 + IsMRKQg*-8.327919006347656 + AHPROQg*0.280586481094360	0.8681795	2	31
122	23167_	y=4.057913303375244 + imDrkQt*-4.935928344726563 + LHDROQg*0.097806222736835 + aSPRtQg*0.059412382543087	0.938799143	3	31
123	23167_	y=4.060825824737549 + imDrkQt*-4.952915191650390 + LHDROQg*0.097219981253147 + iSPRtQg*0.059157751500607	0.939431727	3	31
124	408461_	y=1.742758248357947 + inPRlQg*0.100852680149301 + lPDMqMg*0.003099560484118	0.80557016	2	40
125	408461_	y=1.139803152319389 + inPRlQg*0.087893430183274 +	0.917551726	4	40

		lPDMoMg*0.003517539776609 + iAMRqQg*2.431579364841151 + inMRkQt*1.038853874815755			
126	408462_	y=-4.447927898225615 + imDdSCg*2.435295979512419 + iiMrqQg*0.094635156307397	0.785285688	2	40
127	408462_	y=-9.985955210032028 + imDdSCg*4.564366234578961 + isDrqQg*0.002945010877990 + IIMDQQg*5.203670331019382 + lmMrsGg*1.483206222543829	0.903722129	4	40
128	408464_	y=0.802847494580493 + inPRlQg*0.111337641528975 + iHMMTQt*0.000000009980546	0.75212455	2	40
129	408464_	y=0.624897025648112 + inPRlQg*0.105048221495900 + iHMMTQt*0.000000009918681 + IHMDTQg*-9.248476294178811 + InPdJQg*1.727457275576125	0.920210241	4	40
130	Dipeptides_	y=2.567506668233360 + ibDMFHt*-1.471170216670452 + ISPdlMg*0.117473661385305	0.84794137	2	58
131	Dipeptides_	y=4.581803041430724 + IHMdKEg*-0.039915577766273 + IBPmpGg*0.639122246849638	0.849074656	2	58
132	Dipeptides_	y=2.208486080169678 + IbMmjHg*0.248430266976357 + IbPdPHg*0.020541135221720 + IBMRQCg*-0.265679538249969	0.879454732	3	58
133	Dipeptides_	y=15.605128288269043 + ibDMFHt*-1.721762657165527 + ISPdlMg*0.124963581562042 + imDmFEt*-166.574951171875008 + ImPrSEt*-0.121208801865578	0.903556645	4	58

134	Dipeptides_	y=-7.197427749633789 + IbMmjHg*0.236571803689003 + IbPdPHg*0.020058955997229 + IBMRQCg*-0.236558765172958 + ImDmEEt*2.080404758453369 + ImDrFEt*-0.038597311824560	0.925133169	5	58
135	PCB_lkow_	y=3.120874910401321 + IIDDKGg*-0.441475190602858 + IHDRKEg*0.044664834835730	0.889745045	2	206
136	PCB_lkow_	y=3.039301324572242 + IIDDKGg*-0.420795249377539 + IHDRKEg*0.044187988743215 + aHMmjQt*0.069692154874390 + aSMMjQg*-37.502291360990682	0.916807642	4	206
137	PCB_lkow_	y=-12.993183460296381 + aHMmjQt*0.073027271465754 + aSMMjQg*-41.685504895336179 + iBMmwHg*1195.953967146027040 + iBMmwHg*1195.781250000000000 + iFPMECg*0.000000000046040 + inPRjQt*-0.053628481878679	0.92728403	6	206
138	PCB_rrf_	y=5.085278949837590 + imMrFHt*-357.296321414089600 + iHDdFHg*2.156138113196185	0.692921683	2	209
139	PCB_rrf_	y=6.055340736461524 + imMrFHt*-416.942003738513088 + iHDdFHg*2.313956389789302 + iMMMjQg*1.829475297508006 + iAMrVQg*-0.002506945942322	0.736793462	4	209
140	Triazines_	y=5.522010734067013 + iSMMWHg*-8112.253036635951360 + iSMmEQt*194.350344691394656	0.975281158	2	30
141	Triazines_	y=1.741930263231576 + iSMMWHg*-9261.099477423027200 +	0.983012783	3	30

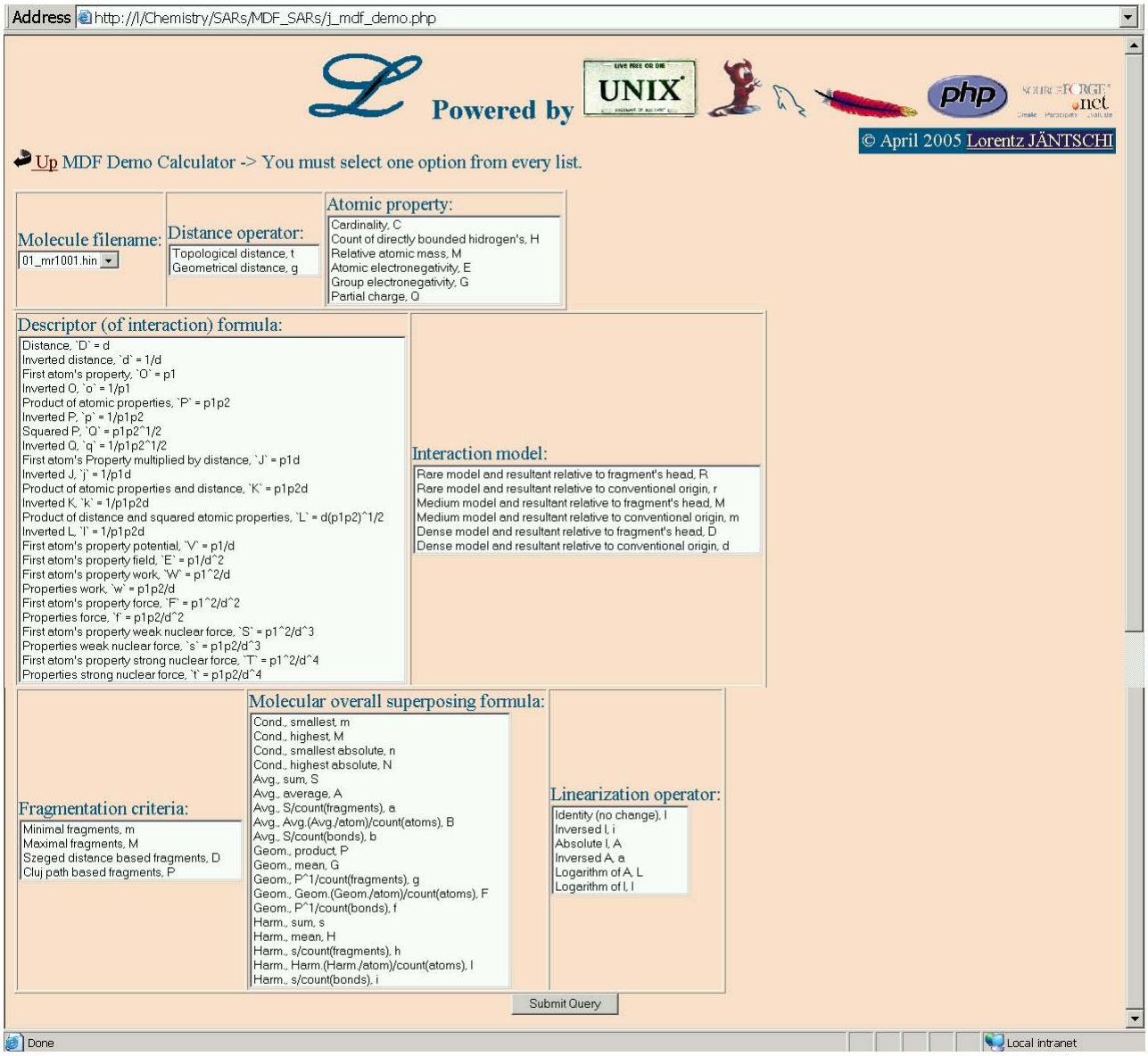
		iAMdEHg*10.338581077497056 + INDRLQg*3.891633816915113			
142	Triazines_	y=5.660616397857666 + iSMmEQt*200.968338012695296 + iSMMWHg*-9010.562500000001280 + LHmrPQg*0.060792036354542 + INPRJQg*2.838208675384522	0.988145828	4	30
143	Triazines_	y=5.753315448760986 + iSMmEQt*198.759780883789088 + iSMMWHg*-9006.287109375000320 + LADmkQt*-0.071008183062077 + INPRJQg*2.863457918167114	0.988538325	4	30
144	Triazines_	y=5.974672317504883 + iSMmEQt*197.155532836914080 + iSMMWHg*-9045.324218750000640 + LBDmkQt*-0.069745272397995 + INPRJQg*2.900454759597779	0.98857367	4	30
145	PCB_rrt_	y=-5.989828784974575 + ISDmsHt*0.023998570742623 + lADrtHg*-1.022659619437935	0.997201518	2	209
146	34121_	y=8.908911583895442E + 000 + liPRLCg*5.130496472711478E + 000 + IAPRVQg*-4.006474847883577E + 001	0.530525286	76	2
147	34121_	y=3.184913714944090E + 000 + iAPrWCt*-7.367710068568820E + 001 + imPdlMg*2.743775705121285E + 000	0.539820496	76	2
148	34121_	y=1.394098640420996E + 001 + lBDDPQg*4.455875932852346E + 000 + IbPDPQg*-9.162168698947033E + 000	0.548704368	76	2
149	34121_	y=1.186554369769341E + 001 + imMRjQg*-3.194956703543156E-001 + INMMwQg*2.201541664377901E + 003 + lBDDPQg*3.723218963557573E + 000 + IbPDPQg*-7.692180382738973E + 000	0.67843743	76	4

150	34121_	y=1.121850860583595E + 001 + imMRjQg*-3.744935336220445E-001 + imDDKQg*7.994101026398718E-001 + lBDDPQg*4.758667402431983E + 000 + IbPDPQg*-7.950113079258346E + 000	0.697382796	76	4
151	34121_	y=1.094115134180832E + 001 + imMRjQg*-3.710272139159518E-001 + imPDKQg*8.467292793176487E-001 + lBDDPQg*4.730475798631939E + 000 + IbPDPQg*-7.809163830375249E + 000	0.714731788	76	4
152	JCCS2001_	y=9.353880712342770E-001 + IHMdpMg*2.405448896100415E + 002 + IHMdOMg*-9.877864464810714E-002	0.883683364	47	2
153	JCCS2001_	y=1.031856753982042E + 000 + IHMdoMg*4.817499113970195E + 001 + IHMdOCg*-3.550966400040491E + 000	0.88733564	47	2
154	JCCS2001_	y=2.021528030038608E + 000 + iAPMLHg*-2.119616905782861E + 001 + aAPrwQt*-1.683039388077890E-004 + IHMdoMg*4.587595185744087E + 001 + IHMdOCg*-3.404948579625964E + 000	0.937366326	47	4
155	JCCS2001_	y=2.628600269530879E + 000 + iAPMLHg*-2.670560915091208E + 001 + imDrDQg*-5.893439946409806E + 000 + IHMdpMg*2.156046225910141E + 002 + IHMdOMg*-9.297899627956369E-002	0.940027138	47	4
156	JCCS2001_	y=8.140489476203218E-001 + lmDRsQg*-5.205699955284682E-002 + iAPrtQg*1.837718043971574E-003 + IHMdpMg*2.408973139743459E + 002 + IHMdOMg*-9.638490508381616E-002	0.940331599	47	4

Rezultatele obținute în urma realizării sistemului prezentat au fost încărcate în formatul web oferit de UEFISCSU (care conține o zonă publică și o zonă privată) și pot fi vizualizate la adresa: http://194.102.64.7/CEEXResurseUmane/ (www.resurse-umane-cdi.ro)

Descrierea sistemului realizat

MDF Demo calculator



Aplicația MDF Demo Calculator permite parametrizarea liberă a modelului MDF şi calculează valoarea descriptorului membru al familiei MDF în conformitate cu parametrizarea aleasă. Permite alegerea ca câte 1 singură opțiune din fiecare listă de opțiuni.

Afişarea rezultatului este precedată de expunerea modului în care acesta a fost obținut. Astfel, alegând fiecare ultima opțiune din fiecare listă, programul afişează:

```
POST data: Array
(
    [hin] => 10_mr1010.hin
    [Do] => g
    [Ap] => Q
    [Df] => t
    [Im] => d
    [Fc] => P
    [Sf] => i
    [Lo] => l
)
Molecule's data: m_c Object
(
    [a] => 12
    [b] => 13
    [atom] => Array
       (
           [1] => Array
              (
                 [0] => 1
```

```
            [1] => -
            [2] => C
            [3] => C4
            [4] => -
            [5] => -0.01363325
            [6] => -1.374815
            [7] => -2.525667
            [8] => -1.163645
            [9] => 2
            [10] => 2
            [11] => 1
            [12] => 6
            [13] => 1
        )

    [2] => Array
        (
            [0] => 2
            [1] => -
            [2] => C
            [3] => C4
            [4] => -
            [5] => -0.03494024
            [6] => -0.6541022
            [7] => -3.076405
            [8] => 0.08466685
            [9] => 2
            [10] => 1
            [11] => 1
            [12] => 3
            [13] => 1
        )

    [3] => Array
        (
            [0] => 3
            [1] => -
            [2] => C
            [3] => C4
            [4] => -
            [5] => -0.01286221
            [6] => 0.8641648
            [7] => -2.810718
            [8] => 0.0138855
            [9] => 2
            [10] => 2
            [11] => 1
            [12] => 4
            [13] => 1
        )

    [4] => Array
        (
            [0] => 4
            [1] => -
            [2] => C
            [3] => C4
            [4] => -
            [5] => -0.2916856
            [6] => 1.165918
            [7] => -1.313278
            [8] => 0.1745337
            [9] => 2
            [10] => 3
            [11] => 1
            [12] => 5
            [13] => 1
        )

    [5] => Array
        (
            [0] => 5
            [1] => -
            [2] => P
            [3] => P5
            [4] => -
            [5] => 0.9998846
            [6] => 0.3477604
            [7] => -0.3762538
            [8] => -1.163645
            [9] => 3
            [10] => 4
            [11] => 1
            [12] => 6
            [13] => 1
            [14] => 7
            [15] => 1
        )

    [6] => Array
        (
            [0] => 6
            [1] => -
            [2] => C
            [3] => C4
            [4] => -
            [5] => -0.07605362
            [6] => -1.374815
            [7] => -0.9852762
            [8] => -1.163645
            [9] => 2
            [10] => 5
            [11] => 1
            [12] => 1
            [13] => 1
        )

    [7] => Array
        (
            [0] => 7
            [1] => -
            [2] => C
            [3] => C3
            [4] => -
            [5] => -0.2473259
            [6] => 0.2043564
            [7] => 1.029266
            [8] => -0.0003523827
            [9] => 3
            [10] => 5
            [11] => 1
            [12] => 8
            [13] => 2
            [14] => 9
            [15] => 1
        )

    [8] => Array
```

```
    (
      [0] => 8
      [1] => -
      [2] => C
      [3] => C3
      [4] => -
      [5] => -0.04595566
      [6] => -0.4518933
      [7] => 0.876143
      [8] => 1.157674
      [9] => 2
      [10] => 7
      [11] => 2
      [12] => 10
      [13] => 1
    )

  [9] => Array
    (
      [0] => 9
      [1] => -
      [2] => C
      [3] => C3
      [4] => -
      [5] => -0.04757643
      [6] => 0.7997317
      [7] => 2.319123
      [8] => -0.3369715
      [9] => 2
      [10] => 7
      [11] => 1
      [12] => 11
      [13] => 2
    )

  [10] => Array
    (
      [0] => 10
      [1] => -
      [2] => C
      [3] => C3
      [4] => -
      [5] => -0.04881668
      [6] => -0.5716058
      [7] => 1.999108
      [8] => 2.082884
      [9] => 2
      [10] => 8
      [11] => 1
      [12] => 12
      [13] => 2
    )

  [11] => Array
    (
      [0] => 11
      [1] => -
      [2] => C
      [3] => C3
      [4] => -
      [5] => -0.0406456
      [6] => 0.6907468
      [7] => 3.349581
      [8] => 0.5127543
      [9] => 2
      [10] => 9
      [11] => 2
      [12] => 12
      [13] => 1
    )

  [12] => Array
    (
      [0] => 12
      [1] => -
      [2] => C
      [3] => C3
      [4] => -
      [5] => -0.05546284
      [6] => -0.02339876
      [7] => 3.182384
      [8] => 1.775193
      [9] => 2
      [10] => 10
      [11] => 2
      [12] => 11
      [13] => 1
    )

)

[prop] => Array
  (
    [1] => Array
      (
        [0] => 1
        [1] => 3
        [2] => 12
        [3] => 2.746
        [4] => 2.64234898371313
        [5] => -0.01363325
      )

    [2] => Array
      (
        [0] => 1
        [1] => 3
        [2] => 12
        [3] => 2.746
        [4] => 2.64234898371313
        [5] => -0.03494024
      )

    [3] => Array
      (
        [0] => 1
        [1] => 3
        [2] => 12
        [3] => 2.746
        [4] => 2.64234898371313
        [5] => -0.01286221
      )

    [4] => Array
      (
        [0] => 1
        [1] => 3
        [2] => 12
```

```
            [3] => 2.746
            [4] => 2.64234898371313
            [5] => -0.2916856
        )

    [5] => Array
        (
            [0] => 1
            [1] => 1
            [2] => 30.9737634
            [3] => 2.515
            [4] => 2.515
            [5] => 0.9998846
        )

    [6] => Array
        (
            [0] => 1
            [1] => 3
            [2] => 12
            [3] => 2.746
            [4] => 2.64234898371313
            [5] => -0.07605362
        )

    [7] => Array
        (
            [0] => 1
            [1] => 1
            [2] => 12
            [3] => 2.746
            [4] => 2.746
            [5] => -0.2473259
        )

    [8] => Array
        (
            [0] => 1
            [1] => 2
            [2] => 12
            [3] => 2.746
            [4] => 2.66788905316544
            [5] => -0.04595566
        )

    [9] => Array
        (
            [0] => 1
            [1] => 2
            [2] => 12
            [3] => 2.746
            [4] => 2.66788905316544
            [5] => -0.04757643
        )

    [10] => Array
        (
            [0] => 1
            [1] => 2
            [2] => 12
            [3] => 2.746
            [4] => 2.66788905316544
            [5] => -0.04881668
        )

    [11] => Array
        (
            [0] => 1
            [1] => 2
            [2] => 12
            [3] => 2.746
            [4] => 2.66788905316544
            [5] => -0.0406456
        )

    [12] => Array
        (
            [0] => 1
            [1] => 2
            [2] => 12
            [3] => 2.746
            [4] => 2.66788905316544
            [5] => -0.05546284
        )

)
    [e] => seed 0
    [f] => forcefield mm+
    [m] => mol 1
    [s] => sys 0 0 1
)
Fragments tree structure: Array
(
    [1] => Array
        (
            [2] => Array
                (
                    [0] => Array
                        (
                            [0] => 1
                            [1] => 5
                            [2] => 6
                            [3] => 7
                            [4] => 8
                            [5] => 9
                            [6] => 10
                            [7] => 11
                            [8] => 12
                        )

                )

            [3] => Array
                (
                    [0] => Array
                        (
                            [0] => 1
                            [1] => 6
                        )

                )

            [4] => Array
                (
                    [0] => Array
                        (
                            [0] => 1
```

```
            [1] => 6
        )

    [1] => Array
        (
            [0] => 1
            [1] => 2
        )

)

[5] => Array
(
    [0] => Array
        (
            [0] => 1
            [1] => 2
        )

)

[6] => Array
(
    [0] => Array
        (
            [0] => 1
            [1] => 2
            [2] => 3
        )

)

[7] => Array
(
    [0] => Array
        (
            [0] => 1
            [1] => 2
            [2] => 3
            [3] => 4
            [4] => 5
        )

)

[8] => Array
(
    [0] => Array
        (
            [0] => 1
            [1] => 2
            [2] => 3
            [3] => 4
            [4] => 5
        )

)

[9] => Array
(
    [0] => Array
        (
            [0] => 1
            [1] => 2
            [2] => 3
            [3] => 4
            [4] => 5
        )

)

[10] => Array
(
    [0] => Array
        (
            [0] => 1
            [1] => 2
            [2] => 3
            [3] => 4
            [4] => 5
        )

)

[11] => Array
(
    [0] => Array
        (
            [0] => 1
            [1] => 2
            [2] => 3
            [3] => 4
            [4] => 5
        )

)

[12] => Array
(
    [0] => Array
        (
            [0] => 1
            [1] => 2
            [2] => 3
            [3] => 4
            [4] => 5
        )

    [1] => Array
        (
            [0] => 1
            [1] => 2
            [2] => 3
            [3] => 4
            [4] => 5
        )

)

)

[2] => Array
(
    [1] => Array
        (
            [0] => Array
                (
                    [0] => 2
```

```
            [1] => 3
            [2] => 4
        )

    )

[3] => Array
    (
        [0] => Array
        (
            [0] => 2
            [1] => 1
            [2] => 6
        )

    )

[4] => Array
    (
        [0] => Array
        (
            [0] => 2
            [1] => 1
        )

    )

[5] => Array
    (
        [0] => Array
        (
            [0] => 2
            [1] => 3
        )

        [1] => Array
        (
            [0] => 2
            [1] => 1
        )

    )

[6] => Array
    (
        [0] => Array
        (
            [0] => 2
            [1] => 3
        )

    )

[7] => Array
    (
        [0] => Array
        (
            [0] => 2
            [1] => 3
            [2] => 4
            [3] => 5
        )

        [1] => Array
        (
            [0] => 2
            [1] => 1
            [2] => 5
            [3] => 6
        )

    )

[8] => Array
    (
        [0] => Array
        (
            [0] => 2
            [1] => 3
            [2] => 4
            [3] => 5
        )

        [1] => Array
        (
            [0] => 2
            [1] => 1
            [2] => 5
            [3] => 6
        )

    )

[9] => Array
    (
        [0] => Array
        (
            [0] => 2
            [1] => 3
            [2] => 4
            [3] => 5
        )

        [1] => Array
        (
            [0] => 2
            [1] => 1
            [2] => 5
            [3] => 6
        )

    )

[10] => Array
    (
        [0] => Array
        (
            [0] => 2
            [1] => 3
            [2] => 4
            [3] => 5
        )

        [1] => Array
        (
            [0] => 2
            [1] => 1
            [2] => 5
```

```
            [3] => 6
        )

    )

    [11] => Array
    (
        [0] => Array
        (
            [0] => 2
            [1] => 3
            [2] => 4
            [3] => 5
        )

        [1] => Array
        (
            [0] => 2
            [1] => 1
            [2] => 5
            [3] => 6
        )

    )

    [12] => Array
    (
        [0] => Array
        (
            [0] => 2
            [1] => 3
            [2] => 4
            [3] => 5
        )

        [1] => Array
        (
            [0] => 2
            [1] => 3
            [2] => 4
            [3] => 5
        )

        [2] => Array
        (
            [0] => 2
            [1] => 1
            [2] => 5
            [3] => 6
        )

        [3] => Array
        (
            [0] => 2
            [1] => 1
            [2] => 5
            [3] => 6
        )

    )

)

[3] => Array
(
    [1] => Array
    (
        [0] => Array
        (
            [0] => 3
            [1] => 4
        )

    )

    [2] => Array
    (
        [0] => Array
        (
            [0] => 3
            [1] => 4
            [2] => 5
            [3] => 7
            [4] => 8
            [5] => 9
            [6] => 10
            [7] => 11
            [8] => 12
        )

    )

    [4] => Array
    (
        [0] => Array
        (
            [0] => 3
            [1] => 1
            [2] => 2
        )

    )

    [5] => Array
    (
        [0] => Array
        (
            [0] => 3
            [1] => 2
        )

    )

    [6] => Array
    (
        [0] => Array
        (
            [0] => 3
            [1] => 4
        )

        [1] => Array
        (
            [0] => 3
            [1] => 2
        )

    )
```

```
            [7] => Array
               (
                 [0] => Array
                  (
                      [0] => 3
                      [1] => 1
                      [2] => 2
                      [3] => 5
                      [4] => 6
                  )

               )

            [8] => Array
               (
                 [0] => Array
                  (
                      [0] => 3
                      [1] => 1
                      [2] => 2
                      [3] => 5
                      [4] => 6
                  )

               )

            [9] => Array
               (
                 [0] => Array
                  (
                      [0] => 3
                      [1] => 1
                      [2] => 2
                      [3] => 5
                      [4] => 6
                  )

               )

            [10] => Array
               (
                 [0] => Array
                  (
                      [0] => 3
                      [1] => 1
                      [2] => 2
                      [3] => 5
                      [4] => 6
                  )

               )

            [11] => Array
               (
                 [0] => Array
                  (
                      [0] => 3
                      [1] => 1
                      [2] => 2
                      [3] => 5
                      [4] => 6
                  )
```

```
               )

            [12] => Array
               (
                 [0] => Array
                  (
                      [0] => 3
                      [1] => 1
                      [2] => 2
                      [3] => 5
                      [4] => 6
                  )

                 [1] => Array
                  (
                      [0] => 3
                      [1] => 1
                      [2] => 2
                      [3] => 5
                      [4] => 6
                  )

               )

         )

      [4] => Array
         (
            [1] => Array
               (
                 [0] => Array
                  (
                      [0] => 4
                      [1] => 5
                      [2] => 7
                      [3] => 8
                      [4] => 9
                      [5] => 10
                      [6] => 11
                      [7] => 12
                  )

                 [1] => Array
                  (
                      [0] => 4
                      [1] => 3
                  )

               )

            [2] => Array
               (
                 [0] => Array
                  (
                      [0] => 4
                      [1] => 5
                      [2] => 7
                      [3] => 8
                      [4] => 9
                      [5] => 10
                      [6] => 11
                      [7] => 12
                  )
```

```
   )

[3] => Array
   (
     [0] => Array
      (
          [0] => 4
          [1] => 5
          [2] => 6
          [3] => 7
          [4] => 8
          [5] => 9
          [6] => 10
          [7] => 11
          [8] => 12
      )

   )

[5] => Array
   (
     [0] => Array
      (
          [0] => 4
          [1] => 2
          [2] => 3
      )

   )

[6] => Array
   (
     [0] => Array
      (
          [0] => 4
          [1] => 3
      )

   )

[7] => Array
   (
     [0] => Array
      (
          [0] => 4
          [1] => 1
          [2] => 2
          [3] => 3
          [4] => 5
          [5] => 6
      )

   )

[8] => Array
   (
     [0] => Array
      (
          [0] => 4
          [1] => 1
          [2] => 2
          [3] => 3
          [4] => 5
          [5] => 6
      )

   )

[9] => Array
   (
     [0] => Array
      (
          [0] => 4
          [1] => 1
          [2] => 2
          [3] => 3
          [4] => 5
          [5] => 6
      )

   )

[10] => Array
   (
     [0] => Array
      (
          [0] => 4
          [1] => 1
          [2] => 2
          [3] => 3
          [4] => 5
          [5] => 6
      )

   )

[11] => Array
   (
     [0] => Array
      (
          [0] => 4
          [1] => 1
          [2] => 2
          [3] => 3
          [4] => 5
          [5] => 6
      )

   )

[12] => Array
   (
     [0] => Array
      (
          [0] => 4
          [1] => 1
          [2] => 2
          [3] => 3
          [4] => 5
          [5] => 6
      )

     [1] => Array
      (
          [0] => 4
          [1] => 1
          [2] => 2
          [3] => 3
```

```
                    [4] => 5
                    [5] => 6
                )

            )

    )

[5] => Array
    (
        [1] => Array
            (
                [0] => Array
                (
                    [0] => 5
                    [1] => 4
                    [2] => 7
                    [3] => 8
                    [4] => 9
                    [5] => 10
                    [6] => 11
                    [7] => 12
                )

            )

        [2] => Array
            (
                [0] => Array
                (
                    [0] => 5
                    [1] => 4
                    [2] => 7
                    [3] => 8
                    [4] => 9
                    [5] => 10
                    [6] => 11
                    [7] => 12
                )

                [1] => Array
                (
                    [0] => 5
                    [1] => 6
                    [2] => 7
                    [3] => 8
                    [4] => 9
                    [5] => 10
                    [6] => 11
                    [7] => 12
                )

            )

        [3] => Array
            (
                [0] => Array
                (
                    [0] => 5
                    [1] => 6
                    [2] => 7
                    [3] => 8
                    [4] => 9
                    [5] => 10
                    [6] => 11
                    [7] => 12
                )

            )

        [4] => Array
            (
                [0] => Array
                (
                    [0] => 5
                    [1] => 1
                    [2] => 6
                    [3] => 7
                    [4] => 8
                    [5] => 9
                    [6] => 10
                    [7] => 11
                    [8] => 12
                )

            )

        [6] => Array
            (
                [0] => Array
                (
                    [0] => 5
                    [1] => 3
                    [2] => 4
                    [3] => 7
                    [4] => 8
                    [5] => 9
                    [6] => 10
                    [7] => 11
                    [8] => 12
                )

            )

        [7] => Array
            (
                [0] => Array
                (
                    [0] => 5
                    [1] => 1
                    [2] => 2
                    [3] => 3
                    [4] => 4
                    [5] => 6
                )

            )

        [8] => Array
            (
                [0] => Array
                (
                    [0] => 5
                    [1] => 1
                    [2] => 2
                    [3] => 3
                    [4] => 4
                    [5] => 6
```

```
      )

   )

[9] => Array
   (
     [0] => Array
      (
          [0] => 5
          [1] => 1
          [2] => 2
          [3] => 3
          [4] => 4
          [5] => 6
      )

   )

[10] => Array
   (
     [0] => Array
      (
          [0] => 5
          [1] => 1
          [2] => 2
          [3] => 3
          [4] => 4
          [5] => 6
      )

   )

[11] => Array
   (
     [0] => Array
      (
          [0] => 5
          [1] => 1
          [2] => 2
          [3] => 3
          [4] => 4
          [5] => 6
      )

   )

[12] => Array
   (
     [0] => Array
      (
          [0] => 5
          [1] => 1
          [2] => 2
          [3] => 3
          [4] => 4
          [5] => 6
      )

     [1] => Array
      (
          [0] => 5
          [1] => 1
          [2] => 2
          [3] => 3
          [4] => 4
          [5] => 6
      )

   )

)

[6] => Array
   (
     [1] => Array
        (
          [0] => Array
           (
               [0] => 6
               [1] => 4
               [2] => 5
               [3] => 7
               [4] => 8
               [5] => 9
               [6] => 10
               [7] => 11
               [8] => 12
           )

        )

     [2] => Array
        (
          [0] => Array
           (
               [0] => 6
               [1] => 5
               [2] => 7
               [3] => 8
               [4] => 9
               [5] => 10
               [6] => 11
               [7] => 12
           )

        )

     [3] => Array
        (
          [0] => Array
           (
               [0] => 6
               [1] => 5
               [2] => 7
               [3] => 8
               [4] => 9
               [5] => 10
               [6] => 11
               [7] => 12
           )

          [1] => Array
           (
               [0] => 6
               [1] => 1
           )

        )
```

```
            [4] => Array
               (
                 [0] => Array
                  (
                      [0] => 6
                      [1] => 1
                  )

               )

            [5] => Array
               (
                 [0] => Array
                  (
                      [0] => 6
                      [1] => 1
                      [2] => 2
                  )

               )

            [7] => Array
               (
                 [0] => Array
                  (
                      [0] => 6
                      [1] => 1
                      [2] => 2
                      [3] => 3
                      [4] => 4
                      [5] => 5
                  )

               )

            [8] => Array
               (
                 [0] => Array
                  (
                      [0] => 6
                      [1] => 1
                      [2] => 2
                      [3] => 3
                      [4] => 4
                      [5] => 5
                  )

               )

            [9] => Array
               (
                 [0] => Array
                  (
                      [0] => 6
                      [1] => 1
                      [2] => 2
                      [3] => 3
                      [4] => 4
                      [5] => 5
                  )

               )
```

```
            [10] => Array
               (
                 [0] => Array
                  (
                      [0] => 6
                      [1] => 1
                      [2] => 2
                      [3] => 3
                      [4] => 4
                      [5] => 5
                  )

               )

            [11] => Array
               (
                 [0] => Array
                  (
                      [0] => 6
                      [1] => 1
                      [2] => 2
                      [3] => 3
                      [4] => 4
                      [5] => 5
                  )

               )

            [12] => Array
               (
                 [0] => Array
                  (
                      [0] => 6
                      [1] => 1
                      [2] => 2
                      [3] => 3
                      [4] => 4
                      [5] => 5
                  )

                 [1] => Array
                  (
                      [0] => 6
                      [1] => 1
                      [2] => 2
                      [3] => 3
                      [4] => 4
                      [5] => 5
                  )

               )

         )

      [7] => Array
         (
            [1] => Array
               (
                 [0] => Array
                  (
                      [0] => 7
                      [1] => 8
                      [2] => 9
                      [3] => 10
```

```
            [4] => 11
            [5] => 12
        )

    )

[2] => Array
    (
      [0] => Array
        (
            [0] => 7
            [1] => 8
            [2] => 9
            [3] => 10
            [4] => 11
            [5] => 12
        )

      [1] => Array
        (
            [0] => 7
            [1] => 8
            [2] => 9
            [3] => 10
            [4] => 11
            [5] => 12
        )

    )

[3] => Array
    (
      [0] => Array
        (
            [0] => 7
            [1] => 8
            [2] => 9
            [3] => 10
            [4] => 11
            [5] => 12
        )

    )

[4] => Array
    (
      [0] => Array
        (
            [0] => 7
            [1] => 8
            [2] => 9
            [3] => 10
            [4] => 11
            [5] => 12
        )

    )

[5] => Array
    (
      [0] => Array
        (
            [0] => 7
            [1] => 8
            [2] => 9
            [3] => 10
            [4] => 11
            [5] => 12
        )

    )

[6] => Array
    (
      [0] => Array
        (
            [0] => 7
            [1] => 8
            [2] => 9
            [3] => 10
            [4] => 11
            [5] => 12
        )

    )

[8] => Array
    (
      [0] => Array
        (
            [0] => 7
            [1] => 1
            [2] => 2
            [3] => 3
            [4] => 4
            [5] => 5
            [6] => 6
            [7] => 9
            [8] => 11
        )

    )

[9] => Array
    (
      [0] => Array
        (
            [0] => 7
            [1] => 1
            [2] => 2
            [3] => 3
            [4] => 4
            [5] => 5
            [6] => 6
            [7] => 8
            [8] => 10
        )

    )

[10] => Array
    (
      [0] => Array
        (
            [0] => 7
            [1] => 1
            [2] => 2
            [3] => 3
```

```
                    [4] => 4
                    [5] => 5
                    [6] => 6
                    [7] => 9
                )

            )

        [11] => Array
            (
                [0] => Array
                (
                    [0] => 7
                    [1] => 1
                    [2] => 2
                    [3] => 3
                    [4] => 4
                    [5] => 5
                    [6] => 6
                    [7] => 8
                )

            )

        [12] => Array
            (
                [0] => Array
                (
                    [0] => 7
                    [1] => 1
                    [2] => 2
                    [3] => 3
                    [4] => 4
                    [5] => 5
                    [6] => 6
                    [7] => 9
                )

                [1] => Array
                (
                    [0] => 7
                    [1] => 1
                    [2] => 2
                    [3] => 3
                    [4] => 4
                    [5] => 5
                    [6] => 6
                    [7] => 8
                )

            )

    )

[8] => Array
    (
        [1] => Array
            (
                [0] => Array
                (
                    [0] => 8
                    [1] => 7
                    [2] => 9
                    [3] => 10
                    [4] => 11
                    [5] => 12
                )

            )

        [2] => Array
            (
                [0] => Array
                (
                    [0] => 8
                    [1] => 7
                    [2] => 9
                    [3] => 10
                    [4] => 11
                    [5] => 12
                )

                [1] => Array
                (
                    [0] => 8
                    [1] => 7
                    [2] => 9
                    [3] => 10
                    [4] => 11
                    [5] => 12
                )

            )

        [3] => Array
            (
                [0] => Array
                (
                    [0] => 8
                    [1] => 7
                    [2] => 9
                    [3] => 10
                    [4] => 11
                    [5] => 12
                )

            )

        [4] => Array
            (
                [0] => Array
                (
                    [0] => 8
                    [1] => 7
                    [2] => 9
                    [3] => 10
                    [4] => 11
                    [5] => 12
                )

            )

        [5] => Array
            (
                [0] => Array
                (
                    [0] => 8
                    [1] => 7
```

```
            [2] => 9
            [3] => 10
            [4] => 11
            [5] => 12
        )

    )

[6] => Array
    (
      [0] => Array
        (
            [0] => 8
            [1] => 7
            [2] => 9
            [3] => 10
            [4] => 11
            [5] => 12
        )

    )

[7] => Array
    (
      [0] => Array
        (
            [0] => 8
            [1] => 10
            [2] => 12
        )

    )

[9] => Array
    (
      [0] => Array
        (
            [0] => 8
            [1] => 10
        )

    )

[10] => Array
    (
      [0] => Array
        (
            [0] => 8
            [1] => 1
            [2] => 2
            [3] => 3
            [4] => 4
            [5] => 5
            [6] => 6
            [7] => 7
            [8] => 9
        )

    )

[11] => Array
    (
      [0] => Array
        (
```

```
            [0] => 8
            [1] => 10
        )

      [1] => Array
        (
            [0] => 8
            [1] => 1
            [2] => 2
            [3] => 3
            [4] => 4
            [5] => 5
            [6] => 6
            [7] => 7
        )

    )

[12] => Array
    (
      [0] => Array
        (
            [0] => 8
            [1] => 1
            [2] => 2
            [3] => 3
            [4] => 4
            [5] => 5
            [6] => 6
            [7] => 7
        )

    )

)

[9] => Array
    (
      [1] => Array
        (
          [0] => Array
            (
              [0] => 9
              [1] => 7
              [2] => 8
              [3] => 10
              [4] => 11
              [5] => 12
            )

        )

      [2] => Array
        (
          [0] => Array
            (
              [0] => 9
              [1] => 7
              [2] => 8
              [3] => 10
              [4] => 11
              [5] => 12
            )
```

```
        [1] => Array
         (
             [0] => 9
             [1] => 7
             [2] => 8
             [3] => 10
             [4] => 11
             [5] => 12
         )

      )

   [3] => Array
      (
        [0] => Array
         (
             [0] => 9
             [1] => 7
             [2] => 8
             [3] => 10
             [4] => 11
             [5] => 12
         )

      )

   [4] => Array
      (
        [0] => Array
         (
             [0] => 9
             [1] => 7
             [2] => 8
             [3] => 10
             [4] => 11
             [5] => 12
         )

      )

   [5] => Array
      (
        [0] => Array
         (
             [0] => 9
             [1] => 7
             [2] => 8
             [3] => 10
             [4] => 11
             [5] => 12
         )

      )

   [6] => Array
      (
        [0] => Array
         (
             [0] => 9
             [1] => 7
             [2] => 8
             [3] => 10
             [4] => 11
             [5] => 12
         )

      )

   [7] => Array
      (
        [0] => Array
         (
             [0] => 9
             [1] => 11
             [2] => 12
         )

      )

   [8] => Array
      (
        [0] => Array
         (
             [0] => 9
             [1] => 11
         )

      )

   [10] => Array
      (
        [0] => Array
         (
             [0] => 9
             [1] => 11
         )

        [1] => Array
         (
             [0] => 9
             [1] => 1
             [2] => 2
             [3] => 3
             [4] => 4
             [5] => 5
             [6] => 6
             [7] => 7
         )

      )

   [11] => Array
      (
        [0] => Array
         (
             [0] => 9
             [1] => 1
             [2] => 2
             [3] => 3
             [4] => 4
             [5] => 5
             [6] => 6
             [7] => 7
             [8] => 8
         )

      )
```

```
        [12] => Array
           (
             [0] => Array
              (
                  [0] => 9
                  [1] => 1
                  [2] => 2
                  [3] => 3
                  [4] => 4
                  [5] => 5
                  [6] => 6
                  [7] => 7
              )

           )

    )

[10] => Array
    (
        [1] => Array
           (
             [0] => Array
              (
                  [0] => 10
                  [1] => 7
                  [2] => 9
                  [3] => 11
                  [4] => 12
              )

           )

        [2] => Array
           (
             [0] => Array
              (
                  [0] => 10
                  [1] => 7
                  [2] => 9
                  [3] => 11
                  [4] => 12
              )

             [1] => Array
              (
                  [0] => 10
                  [1] => 7
                  [2] => 9
                  [3] => 11
                  [4] => 12
              )

           )

        [3] => Array
           (
             [0] => Array
              (
                  [0] => 10
                  [1] => 7
                  [2] => 9
                  [3] => 11
                  [4] => 12
```

```
              )

           )

        [4] => Array
           (
             [0] => Array
              (
                  [0] => 10
                  [1] => 7
                  [2] => 9
                  [3] => 11
                  [4] => 12
              )

           )

        [5] => Array
           (
             [0] => Array
              (
                  [0] => 10
                  [1] => 7
                  [2] => 9
                  [3] => 11
                  [4] => 12
              )

           )

        [6] => Array
           (
             [0] => Array
              (
                  [0] => 10
                  [1] => 7
                  [2] => 9
                  [3] => 11
                  [4] => 12
              )

           )

        [7] => Array
           (
             [0] => Array
              (
                  [0] => 10
                  [1] => 12
              )

           )

        [8] => Array
           (
             [0] => Array
              (
                  [0] => 10
                  [1] => 11
                  [2] => 12
              )

           )
```

```
            [9] => Array
               (
                 [0] => Array
                  (
                      [0] => 10
                      [1] => 12
                  )

                 [1] => Array
                  (
                      [0] => 10
                      [1] => 8
                  )

               )

            [11] => Array
               (
                 [0] => Array
                  (
                      [0] => 10
                      [1] => 8
                  )

               )

            [12] => Array
               (
                 [0] => Array
                  (
                      [0] => 10
                      [1] => 1
                      [2] => 2
                      [3] => 3
                      [4] => 4
                      [5] => 5
                      [6] => 6
                      [7] => 7
                      [8] => 8
                  )

               )

         )

      [11] => Array
         (
            [1] => Array
               (
                 [0] => Array
                  (
                      [0] => 11
                      [1] => 7
                      [2] => 8
                      [3] => 10
                      [4] => 12
                  )

               )

            [2] => Array
               (
                 [0] => Array
                  (
                      [0] => 11
                      [1] => 7
                      [2] => 8
                      [3] => 10
                      [4] => 12
                  )

                [1] => Array
                  (
                      [0] => 11
                      [1] => 7
                      [2] => 8
                      [3] => 10
                      [4] => 12
                  )

               )

            [3] => Array
               (
                 [0] => Array
                  (
                      [0] => 11
                      [1] => 7
                      [2] => 8
                      [3] => 10
                      [4] => 12
                  )

               )

            [4] => Array
               (
                 [0] => Array
                  (
                      [0] => 11
                      [1] => 7
                      [2] => 8
                      [3] => 10
                      [4] => 12
                  )

               )

            [5] => Array
               (
                 [0] => Array
                  (
                      [0] => 11
                      [1] => 7
                      [2] => 8
                      [3] => 10
                      [4] => 12
                  )

               )

            [6] => Array
               (
                 [0] => Array
                  (
                      [0] => 11
                      [1] => 7
                      [2] => 8
```

```
            [3] => 10
            [4] => 12
        )

    )

[7] => Array
    (
        [0] => Array
            (
                [0] => 11
                [1] => 12
            )

    )

[8] => Array
    (
        [0] => Array
            (
                [0] => 11
                [1] => 12
            )

        [1] => Array
            (
                [0] => 11
                [1] => 9
            )

    )

[9] => Array
    (
        [0] => Array
            (
                [0] => 11
                [1] => 10
                [2] => 12
            )

    )

[10] => Array
    (
        [0] => Array
            (
                [0] => 11
                [1] => 9
            )

    )

[12] => Array
    (
        [0] => Array
            (
                [0] => 11
                [1] => 1
                [2] => 2
                [3] => 3
                [4] => 4
                [5] => 5
                [6] => 6
                [7] => 7
                [8] => 9
            )

        )

    )

[12] => Array
    (
        [1] => Array
            (
                [0] => Array
                    (
                        [0] => 12
                        [1] => 7
                        [2] => 9
                        [3] => 11
                    )

                [1] => Array
                    (
                        [0] => 12
                        [1] => 7
                        [2] => 8
                        [3] => 10
                    )

            )

        [2] => Array
            (
                [0] => Array
                    (
                        [0] => 12
                        [1] => 7
                        [2] => 9
                        [3] => 11
                    )

                [1] => Array
                    (
                        [0] => 12
                        [1] => 7
                        [2] => 8
                        [3] => 10
                    )

                [2] => Array
                    (
                        [0] => 12
                        [1] => 7
                        [2] => 9
                        [3] => 11
                    )

                [3] => Array
                    (
                        [0] => 12
                        [1] => 7
                        [2] => 8
                        [3] => 10
                    )
```

```
    )

[3] => Array
    (
      [0] => Array
       (
           [0] => 12
           [1] => 7
           [2] => 9
           [3] => 11
       )

      [1] => Array
       (
           [0] => 12
           [1] => 7
           [2] => 8
           [3] => 10
       )

    )

[4] => Array
    (
      [0] => Array
       (
           [0] => 12
           [1] => 7
           [2] => 9
           [3] => 11
       )

      [1] => Array
       (
           [0] => 12
           [1] => 7
           [2] => 8
           [3] => 10
       )

    )

[5] => Array
    (
      [0] => Array
       (
           [0] => 12
           [1] => 7
           [2] => 9
           [3] => 11
       )

      [1] => Array
       (
           [0] => 12
           [1] => 7
           [2] => 8
           [3] => 10
       )

    )

[6] => Array
    (
      [0] => Array
       (
           [0] => 12
           [1] => 7
           [2] => 9
           [3] => 11
       )

      [1] => Array
       (
           [0] => 12
           [1] => 7
           [2] => 8
           [3] => 10
       )

    )

[7] => Array
    (
      [0] => Array
       (
           [0] => 12
           [1] => 11
       )

      [1] => Array
       (
           [0] => 12
           [1] => 10
       )

    )

[8] => Array
    (
      [0] => Array
       (
           [0] => 12
           [1] => 11
       )

    )

[9] => Array
    (
      [0] => Array
       (
           [0] => 12
           [1] => 10
       )

    )

[10] => Array
    (
      [0] => Array
       (
           [0] => 12
           [1] => 9
           [2] => 11
       )

    )
```

```
            [11] => Array
               (
                  [0] => Array
                   (
                       [0] => 12
                       [1] => 8
                       [2] => 10
                   )

               )

        )

)
Fragments: 170
Fragmental properties: Array
(
    [0] => 0.000255360579194503
    [1] => 1.41159157657011E-05
    [2] => 0.000355196368090578
    [3] => 0.000284002582761743
    [4] => 0.000577937418918154
    [5] => 0.000246662556358222
    [6] => 0.0208356422532841
    [7] => 0.000649603849664492
    [8] => 0.000647973371588807
    [9] => 0.000125877890214753
    [10] => 0.00012043515143826
    [11] => 9.3847246028092E-05
    [12] => 9.3847246028092E-05
    [13] => 0.000117933267956123
    [14] => 8.9418676019586E-05
    [15] => 0.000284002582761743
    [16] => 0.000565617697518195
    [17] => 0.000577937418918154
    [18] => 7.30363892947185E-05
    [19] => 0.0208457696901686
    [20] => 0.0217841636537158
    [21] => 0.000651040598279288
    [22] => 0.000761368233598933
    [23] => 0.000648719064960869
    [24] => 0.000700992058467699
    [25] => 0.000126445352383623
    [26] => 0.000157856998587307
    [27] => 0.000120741470641167
    [28] => 0.000138466358371142
    [29] => 9.42234343777831E-05
    [30] => 9.42234343777831E-05
    [31] => 0.000114218201008345
    [32] => 0.000114218201008345
    [33] => 4.58113305355017E-05
    [34] => 0.000227076704344559
    [35] => 0.000917417655984093
    [36] => 0.000565617697518195
    [37] => 0.000326942005182717
    [38] => 7.30363892947185E-05
    [39] => 0.0217735786356716
    [40] => 0.000759802120755046
    [41] => 0.000700186431777914
    [42] => 0.000157233555982487
    [43] => 0.000138131571827764
    [44] => 0.000113805787187047
    [45] => 0.000113805787187047
    [46] => 0.000189546460019776
    [47] => 4.58113305355017E-05
    [48] => 0.00019593122124718
    [49] => 0.00053557784154892
    [50] => 0.0265790330558128
    [51] => 0.000326942005182717
    [52] => 0.0206027092227705
    [53] => 0.000619300314400984
    [54] => 0.000635465569690894
    [55] => 0.000117530874791904
    [56] => 0.000116064293506768
    [57] => 8.87380463946568E-05
    [58] => 8.87380463946568E-05
    [59] => 0.000189546460019776
    [60] => 0.00019593122124718
    [61] => 0.000277151180889542
    [62] => 0.000199692362622065
    [63] => 0.0245874011982098
    [64] => 0.00630693172882427
    [65] => 0.0206027092227705
    [66] => 0.000619300314400984
    [67] => 0.000635465569690895
    [68] => 0.000117530874791904
    [69] => 0.000116064293506768
    [70] => 8.87380463946567E-05
    [71] => 8.87380463946567E-05
    [72] => 0.000114453824229655
    [73] => 0.000277151180889542
    [74] => 0.000199692362622065
    [75] => 1.41159157657011E-05
    [76] => 0.000355196368090578
    [77] => 0.00724819770810717
    [78] => 0.0206027092227705
    [79] => 0.000619300314400984
    [80] => 0.000635465569690894
    [81] => 0.000117530874791904
    [82] => 0.000116064293506768
    [83] => 8.87380463946568E-05
    [84] => 8.87380463946568E-05
    [85] => 1.64044107463081E-05
    [86] => 3.85953670247642E-05
    [87] => 3.85953670247642E-05
    [88] => 1.81590481000686E-05
    [89] => 0.00206951819593353
    [90] => 0.0238280951138962
    [91] => 0.000365913213111772
    [92] => 0.0030750246797182
    [93] => 0.00219253790989545
    [94] => 0.000267192454431302
    [95] => 0.000211562681369456
    [96] => 0.000202833648165518
    [97] => 0.000204360267542443
    [98] => 1.64044107463081E-05
    [99] => 3.85953670247642E-05
    [100] => 3.85953670247642E-05
    [101] => 1.81590481000686E-05
    [102] => 0.00206951819593353
    [103] => 0.0238280951138962
    [104] => 0.000365913213111772
    [105] => 0.00395423159099302
    [106] => 9.74485775303621E-05
    [107] => 0.000728543152193593
    [108] => 8.47691497067905E-05
    [109] => 0.000211562681369456
```

```
    [110] => 0.000204360267542443
    [111] => 1.64044107463081E-05
    [112] => 3.85953670247642E-05
    [113] => 3.85953670247642E-05
    [114] => 1.81590481000686E-05
    [115] => 0.00206951819593353
    [116] => 0.0238280951138962
    [117] => 0.000365913213111772
    [118] => 0.00297508143811235
    [119] => 9.10370679461242E-05
    [120] => 9.23173530098107E-05
    [121] => 0.000267192454431302
    [122] => 0.000765194717484188
    [123] => 0.000202833648165518
    [124] => 1.45106193753692E-05
    [125] => 3.31215788114097E-05
    [126] => 3.31215788114097E-05
    [127] => 1.61176094831455E-05
    [128] => 0.00189774927661049
    [129] => 0.0231079409713801
    [130] => 0.000331931993247684
    [131] => 0.000555221228973406
    [132] => 0.000576980896741426
    [133] => 0.000110892615891543
    [134] => 9.74485775303621E-05
    [135] => 8.47691497067905E-05
    [136] => 0.000988528423632445
    [137] => 1.56408237281553E-05
    [138] => 3.69297637912785E-05
    [139] => 3.69297637912785E-05
    [140] => 1.73036340554639E-05
    [141] => 0.00199834075063783
    [142] => 0.0231888387397002
    [143] => 0.000352853702773038
    [144] => 0.000494695155248354
    [145] => 0.000100817769587227
    [146] => 9.10370679461242E-05
    [147] => 0.000687439906352944
    [148] => 9.23173530098107E-05
    [149] => 0.000643634341919819
    [150] => 1.38940714416988E-05
    [151] => 1.53219896268676E-05
    [152] => 3.13602877791875E-05
    [153] => 3.6172314936578E-05
    [154] => 3.13602877791875E-05
    [155] => 3.6172314936578E-05
    [156] => 1.54520129285205E-05
    [157] => 1.69508563962762E-05
    [158] => 0.00185847516045578
    [159] => 0.00197525903691226
    [160] => 0.0229470947581165
    [161] => 0.0230527304582322
    [162] => 0.000323832626425813
    [163] => 0.000348352613261895
    [164] => 0.000494695155248354
    [165] => 0.000555221228973406
    [166] => 0.000100817769587227
    [167] => 0.000110892615891543
    [168] => 0.00090996666926623
    [169] => 0.000564198384655363
)
iPdtQg = 6.32451624641893E-06
liPdtQg = -11.9710770090553
```

Sursa aplicației MDF Demo este următoarea:

```
<?
$definition_page["auto_index"]=FALSE;
$definition_page["copyright"]=array("April 2005","lori");
include("/usr/home/www/data/definitions.php");
if(!array_key_exists("hin",$_POST)){
      echo("MDF Demo Calculator -> You must select one option from every
list.");
      $a=`ls -1 demo/`;
      $b=explode("\n",$a);
      unset($b[count($b)-1]);
      unset($a);
      echo("<form method='post' action='".$_SERVER["PHP_SELF"]."'>");
      echo("<table border='1'><tr>");
      echo("<td>Molecule filename:<BR><select name='hin'>");
      for($i=0;$i<count($b);$i++){
            echo("<option value='".$b[$i]."'>".$b[$i]);
      }
      echo("</select>");
      echo("<td>Distance operator:<BR><select name='Do' size='2'>");
      echo("<option value='t'>Topological distance, t");
      echo("<option value='g'>Geometrical distance, g");
      echo("</select>");
      echo("<td>Atomic property:<BR><select name='Ap' size='6'>");
      echo("<option value='C'>Cardinality, C");
      echo("<option value='H'>Count of directly bounded hidrogen's, H");
      echo("<option value='M'>Relative atomic mass, M");
      echo("<option value='E'>Atomic electronegativity, E");
      echo("<option value='G'>Group electronegativity, G");
```

```
      echo("<option value='Q'>Partial charge, Q");
      echo("</select>");
      echo("</table><table border='1'><tr>");
      echo("<td>Descriptor (of interaction) formula:<BR><select name='Df'
size='24'>");
      echo("<option value='D'>Distance, `D` = d");
      echo("<option value='d'>Inverted distance, `d` = 1/d");
      echo("<option value='O'>First atom's property, `O` = p1");
      echo("<option value='o'>Inverted O, `o` = 1/p1");
      echo("<option value='P'>Product of atomic properties, `P` = p1p2");
      echo("<option value='p'>Inverted P, `p` = 1/p1p2");
      echo("<option value='Q'>Squared P, `Q` = p1p2^1/2");
      echo("<option value='q'>Inverted Q, `q` = 1/p1p2^1/2");
      echo("<option value='J'>First atom's Property multiplied by distance, `J`
= p1d");
      echo("<option value='j'>Inverted J, `j` = 1/p1d");
      echo("<option value='K'>Product of atomic properties and distance, `K` =
p1p2d");
      echo("<option value='k'>Inverted K, `k` = 1/p1p2d");
      echo("<option value='L'>Product of distance and squared atomic properties,
`L` = d(p1p2)^1/2");
      echo("<option value='l'>Inverted L, `l` = 1/p1p2d");
      echo("<option value='V'>First atom's property potential, `V` = p1/d");
      echo("<option value='E'>First atom's property field, `E` = p1/d^2");
      echo("<option value='W'>First atom's property work, `W` = p1^2/d");
      echo("<option value='w'>Properties work, `w` = p1p2/d");
      echo("<option value='F'>First atom's property force, `F` = p1^2/d^2");
      echo("<option value='f'>Properties force, `f` = p1p2/d^2");
      echo("<option value='S'>First atom's property weak nuclear force, `S` =
p1^2/d^3");
      echo("<option value='s'>Properties weak nuclear force, `s` = p1p2/d^3");
      echo("<option value='T'>First atom's property strong nuclear force, `T` =
p1^2/d^4");
      echo("<option value='t'>Properties strong nuclear force, `t` = p1p2/d^4");
      echo("</select>");
      echo("<td>Interaction model:<BR><select name='Im' size='6'>");
      echo("<option value='R'>Rare model and resultant relative to fragment's
head, R");
      echo("<option value='r'>Rare model and resultant relative to conventional
origin, r");
      echo("<option value='M'>Medium model and resultant relative to fragment's
head, M");
      echo("<option value='m'>Medium model and resultant relative to
conventional origin, m");
      echo("<option value='D'>Dense model and resultant relative to fragment's
head, D");
      echo("<option value='d'>Dense model and resultant relative to conventional
origin, d");
      echo("</select>");
      echo("</table><table border='1'><tr>");
      echo("<td>Fragmentation criteria:<BR><select name='Fc' size='4'>");
      echo("<option value='m'>Minimal fragments, m");
      echo("<option value='M'>Maximal fragments, M");
      echo("<option value='D'>Szeged distance based fragments, D");
      echo("<option value='P'>Cluj path based fragments, P");
      echo("</select>");
      echo("<td>Molecular overall superposing formula:<BR><select name='Sf'
size='19'>");
      echo("<option value='m'>Cond., smallest, m");
      echo("<option value='M'>Cond., highest, M");
      echo("<option value='n'>Cond., smallest absolute, n");
      echo("<option value='N'>Cond., highest absolute, N");
      echo("<option value='S'>Avg., sum, S");
      echo("<option value='A'>Avg., average, A");
```

```
	echo("<option value='a'>Avg., S/count(fragments), a");
	echo("<option value='B'>Avg., Avg.(Avg./atom)/count(atoms), B");
	echo("<option value='b'>Avg., S/count(bonds), b");
	echo("<option value='P'>Geom., product, P");
	echo("<option value='G'>Geom., mean, G");
	echo("<option value='g'>Geom., P^1/count(fragments), g");
	echo("<option value='F'>Geom., Geom.(Geom./atom)/count(atoms), F");
	echo("<option value='f'>Geom., P^1/count(bonds), f");
	echo("<option value='s'>Harm., sum, s");
	echo("<option value='H'>Harm., mean, H");
	echo("<option value='h'>Harm., s/count(fragments), h");
	echo("<option value='I'>Harm., Harm.(Harm./atom)/count(atoms), I");
	echo("<option value='i'>Harm., s/count(bonds), i");
	echo("</select>");
	echo("<td>Linearization operator:<BR><select name='Lo' size='6'>");
	echo("<option value='I'>Identity (no change), I");
	echo("<option value='i'>Inversed I, i");
	echo("<option value='A'>Absolute I, A");
	echo("<option value='a'>Inversed A, a");
	echo("<option value='L'>Logarithm of A, L");
	echo("<option value='l'>Logarithm of I, l");
	echo("</select>");
	echo("</table><center><input type='submit'></center></form>");
}else {
	include 'f_mdf_functions.php';
	class m_c{
		var $a;//number of atoms
		var $b;//number of bonds
		var $atom;//molecule structure
		var $prop;//property structure
		var $dist;//distances structure
		var $e;//seed
		var $f;//forcefield
		var $m;//molecule number
		var $p;//paths structure
		var $s;//sys
		var $v;//view
		function m_c(){
			$this->a=0;$this->b=0;$this->p[0][0][0]=0;//atomul 0, calea 0,
varful 0 e 0
		}
	}
	class fragments{
		var $a;//number of atoms
		var $f;//fragments structure
		var $d;//substructura de distanta
		function fragments(&$mol){//!!!!!!!!!!!!!de implementat cu foreach
			$this->a=$mol->a;
			foreach ($mol->atom as $atom_i){//fiecare varf i
				foreach ($mol->atom as $atom_j){//fiecare varf j
					$this->f[0][$atom_i[0]][$atom_j[0]][0]=$atom_i[0];
				}//primul varf - implicit
				$this->f[0][$atom_i[0]][$atom_i[0]][0]=0;
			}//genereaza fragmentele minimale - ok
		}
	}
	class models{
		var $m_t; var $m_d; var $m_s; var $fty; var $pty; var $dty; var
$smo; var $lty;
		function models(){
			$this->m_t                                                    =
array("mMDP","CHMEGQ","DdOoPpQqJjKkLlVEWwFfSsTt","RrMmDd");
			$this->m_d[0]="pr2m_".strpos($this->m_t[2],$_POST["Df"]);
			$this->m_s[0]="prf_dmr_".strpos($this->m_t[3],$_POST["Im"]);
```

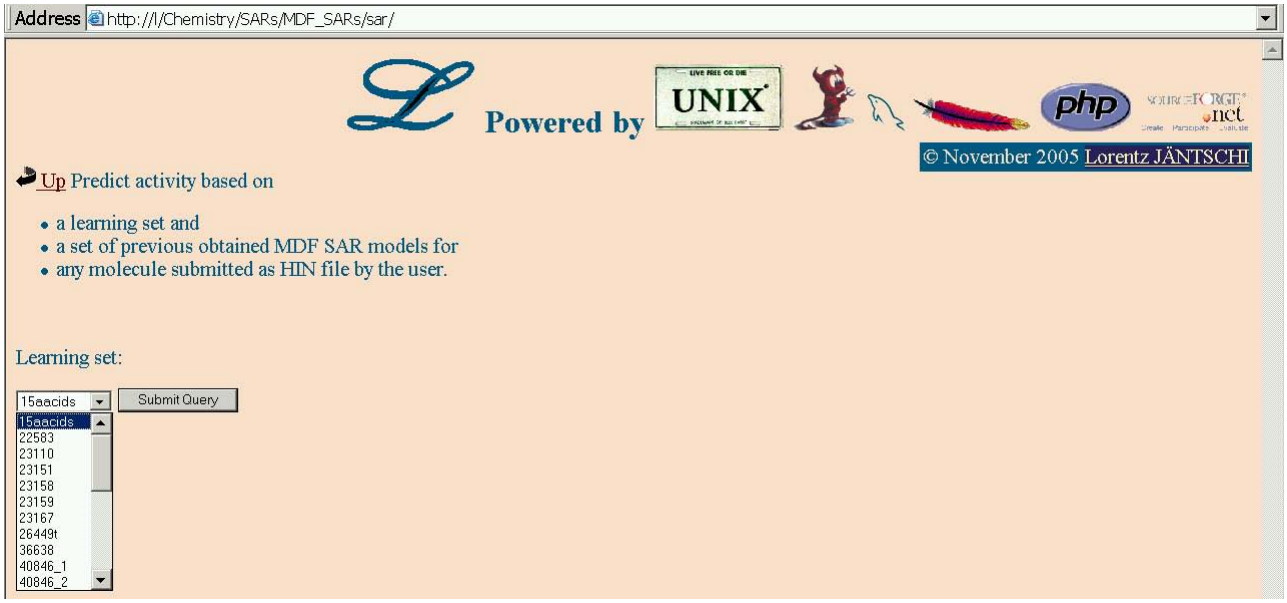
```
                $this->fty=strpos($this->m_t[0],$_POST["Fc"]);
                $this->pty=strpos($this->m_t[1],$_POST["Ap"]);
                $this->dty=strpos("gt",$_POST["Do"]);
                $this->smo=strpos("mMnNSPsAaBbGgFfHhIi",$_POST["Sf"]);
                $this->lty=strpos("IiAaLl",$_POST["Lo"]);
                $this->m_t                                                    =
array($_POST["Fc"],$_POST["Ap"],$_POST["Df"],$_POST["Im"]);
            }
        }
        $b=$_POST["hin"];
        $model = new models;
        $m = new m_c;
        read_mol($m,"demo/".$b,"z_atom_data.txt");
        tree_mol($m,$tv);
        $f = new fragments($m);
        gen_f_max($f,$m);
        gen_f_sdi($f,$m);
        gen_f_fdi($f,$m,$tv);
        unset($f->d);
        $afisa=af_f_pf($f,$m,$model);
        echo("<pre>");
        echo("POST data: ");
        print_r($_POST);
        echo("Molecule's data: ");
        unset($m->dist);
        unset($m->v);
        print_r($m);
        echo("Fragments tree structure: ");
        print_r($f->f[$model->fty]);
        echo($afisa);
        unset($m);
        unset($tv);
        unset($f);
}
function af_f_pf(&$frag,&$mol,&$md){
 $contor=0;
 $tip_f=$md->fty;//tip de fragment (4)
 $tip_p=$md->pty;//tip de proprietate (6)
 $md_d_c=$md->m_d[0];//tip de descriptor (24)
 $md_i_c=$md->m_s[0];//tip de interactiune (6)
 $f_cnt=0;
 foreach ($mol->atom as $atom_i){//fiecare varf i
  $i=$atom_i[0];
  foreach ($mol->atom as $atom_j){//fiecare varf j
   $varf=$atom_j[0];
   if($i==$varf) continue;
   if(!is_array($frag->f[$tip_f][$i][$varf][0]))
    $fragment[0]=$frag->f[$tip_f][$i][$varf];
   else
    $fragment=$frag->f[$tip_f][$i][$varf];
    for($fr_i=0;$fr_i<count($fragment);$fr_i++){
     $tmpp = $md_i_c($tip_p,$md_d_c,$varf,$fragment[$fr_i],$mol);
     $p_array[0][$f_cnt]=$tmpp[0];//gt
     $p_array[1][$f_cnt]=$tmpp[1];
     $f_cnt++;
    }
  }
 }
 $afisa="Fragments: ".$f_cnt."\r\n";
 $afisa.="Fragmental properties: ".print_r($p_array[$md->dty],TRUE);
 $na_fprop="";
 $na_fprop=$md->m_t[0].$md->m_t[3].$md->m_t[2].$md->m_t[1];
 $afisa.=af_cumul($contor,$mol->a,$mol->b,$p_array,$na_fprop,$md->dty,$md-
>smo,$md->lty);
```

```
 unset($p_array);
 unset($frag);
 unset($mol);
 return $afisa;
}//genereaza si afiseaza proprietatile fragmentale
function
af_cumul(&$contor,$nr_at,$nr_bo,&$p_array,&$na_fprop,$dtydty,$smo,$lty){
 $metric="gt";
 $nr_frags=count($p_array[0]);
 for($tip_x=$dtydty;$tip_x<$dtydty+1;$tip_x++){
  $sum=0;
  $prd=1;
  $isu=0;
  $max=-1e100;
  $max_a=0;
  $min=1e100;
  $min_a=1e100;
  $nr_fprop=0;
  for($tip_y=0;$tip_y<$nr_frags;$tip_y++){
   if(!$p_array[$tip_x][$tip_y]) continue;
   if(!is_finite($p_array[$tip_x][$tip_y]))continue;
   $nr_fprop++;
   $sum += $p_array[$tip_x][$tip_y];
   $prd *= $p_array[$tip_x][$tip_y];
   $isu += 1/$p_array[$tip_x][$tip_y];
   if($p_array[$tip_x][$tip_y]>$max) $max=$p_array[$tip_x][$tip_y];
   if(abs($p_array[$tip_x][$tip_y])>$max_a)
$max_a=abs($p_array[$tip_x][$tip_y]);
   if($p_array[$tip_x][$tip_y]<$min) $min=$p_array[$tip_x][$tip_y];
   if(abs($p_array[$tip_x][$tip_y])<$min_a)
$min_a=abs($p_array[$tip_x][$tip_y]);
  }
  if($max==-1e100) $max=(float)"INF";
  $v[0]=$max;//max property
  $af[0]="m".$na_fprop.$metric[$tip_x];
  $v[1]=$max_a;//max_a property
  $af[1]="M".$na_fprop.$metric[$tip_x];
  if($min==1e100) $min=(float)"INF";
  $v[2]=$min;//min property
  $af[2]="n".$na_fprop.$metric[$tip_x];
  if($min_a==1e100) $min_a=(float)"INF";
  $v[3]=$min_a;//min_a property
  $af[3]="N".$na_fprop.$metric[$tip_x];
  $v[4]=$sum;//sum property
  $af[4]="S".$na_fprop.$metric[$tip_x];
  $v[5]=$prd;//product property
  $af[5]="P".$na_fprop.$metric[$tip_x];
  if($isu)
    $v[6]=1/$isu;//harmonic sum property
   else
    $v[6]=(float)"INF";
  $af[6]="s".$na_fprop.$metric[$tip_x];
  if($nr_fprop)
   $v[7]=$sum/$nr_fprop;//average mean property for valid fragments
  else
   $v[7]=(float)"INF";
  $af[7]="A".$na_fprop.$metric[$tip_x];
  $v[8]=$sum/$nr_frags;//average mean property for all fragments
  $af[8]="a".$na_fprop.$metric[$tip_x];
  $v[9]=$sum/$nr_at;//average mean property for all fragments
  $af[9]="B".$na_fprop.$metric[$tip_x];
  $v[10]=$sum/$nr_bo;//average mean property for all fragments
  $af[10]="b".$na_fprop.$metric[$tip_x];
  if($nr_fprop)
```

```
   $v[11]=pow(abs($prd),1/$nr_fprop);//geometric mean property for valid
fragments - without sign
  else
   $v[11]=1;
  $af[11]="G".$na_fprop.$metric[$tip_x];
  $v[12]=pow(abs($prd),1/$nr_frags);//geometric mean property fragments -
without sign
  $af[12]="g".$na_fprop.$metric[$tip_x];
  $v[13]=pow(abs($prd),1/$nr_at);//geometric mean property fragments - without
sign
  $af[13]="F".$na_fprop.$metric[$tip_x];
  $v[14]=pow(abs($prd),1/$nr_bo);//geometric mean property fragments - without
sign
  $af[14]="f".$na_fprop.$metric[$tip_x];
  if($isu)
   $v[15]=$nr_fprop/$isu;//harmonic mean property for valid fragments
  else
   $v[15]=(float)"INF";
  $af[15]="H".$na_fprop.$metric[$tip_x];
  if($isu)
   $v[16]=$nr_frags/$isu;//harmonic mean property for valid fragments
  else
   $v[16]=(float)"INF";
  $af[16]="h".$na_fprop.$metric[$tip_x];
  if($isu)
   $v[17]=$nr_at/$isu;//harmonic mean property for valid fragments
  else
   $v[17]=(float)"INF";
  $af[17]="I".$na_fprop.$metric[$tip_x];
  if($isu)
   $v[18]=$nr_bo/$isu;//harmonic mean property for valid fragments
  else
   $v[18]=(float)"INF";
  $af[18]="i".$na_fprop.$metric[$tip_x];
 }
 $ret = $af[$smo]." = ".$v[$smo]."\r\n";
 if($lty==0){
  $ret .= "I".$af[$smo]." = ".$v[$smo]."\r\n";
 }
 if($lty==1){
  $ret .= "i".$af[$smo]." = ".(1/$v[$smo])."\r\n";
 }
 if($lty==2){
  $ret .= "A".$af[$smo]." = ".(abs($v[$smo]))."\r\n";
 }
 if($lty==3){
  $ret .= "a".$af[$smo]." = ".(abs(1/$v[$smo]))."\r\n";
 }
 if($lty==4){
  $ret .= "L".$af[$smo]." = ".(log(abs($v[$smo])))."\r\n";
 }
 if($lty==5){
  $ret .= "l".$af[$smo]." = ".(log($v[$smo]))."\r\n";
 }
 return $ret;
}
?>
```

MDF SAR analyst

MDF SAR analyst este o aplicație expert care interoghează baza de date cu relații structură-activitate obținute. Captura de ecran a aplicației este redată mai jos:



La apăsarea butonului *Submit Query* se generează o listă cu toate relațiile structură-activitate relevante obținute pe setul ales. Aplicația necesită şi furnizarea unui fişier în format HyperChem care să conțină structura unui compus chimic pentru care se face predicția activității biologice pe baza modelului selectat (figura următoare, alegând ultima opțiune din listă atât pentru imaginea de mai sus cât şi pentru imaginea de mai jos):

Programul afişează activiatea prezisă pe baza modelului:

Molecule file name:

- **30_t30.hin**

Predictor's equation:

- **5.974672317504883+iSMmEQt*197.155532836914080+iSMMWHg*-9045.324218750000640+LBDmkQt*-0.069745272397995+INPRJQg*2.900454759597779**

MDF Members:

- **iSMmEQt = 0.0121378516717032**
- **iSMMWHg = 0.000152405165829042**
- **LBDmkQt = 7.3759602071117**
- **INPRJQg = 0.0593212041385774**

Predicted activity:

- **6.64678290884594**

Sursa programului de predicție este următoarea:

```
<?
$definition_page["up_dir"]="";
$definition_page["auto_index"]=FALSE;
$definition_page["copyright"]=array("November 2005 than March 10, 2007","lori");
include("/usr/home/www/data/definitions.php");
if(!$_POST["equation"])die("no data");
$a=explode("+",$_POST["equation"]);
$coef[0]=$a[0];
$name[0]="";
for($i=1;$i<count($a);$i++){
      list($name[$i],$coef[$i])=explode("*",$a[$i]);
}
if(!isset($_FILES))die("You must provide a file.");
if(!array_key_exists("file",$_FILES))die("You must provide a file.");
$file_name=$_FILES["file"]["name"];
$file_data=@file_get_contents($_FILES["file"]["tmp_name"]);
if(!$file_data)die("You must provide a file.");
if(strpos($file_data,"H              H")>0)die("<pre>\r\n".$file_data."\r\n<font
color='red'><b>You   must   withdraw   Hydrogen's   from   structure   for   MDF
calculation.\r\n</b></font></pre>");
echo("<b>Molecule file name:</b><UL><LI>".$file_name."</UL>");
include("mdf_functions.php");
include("mdf_predictus.php");
class m_c{
 var $a;//number of atoms
 var $b;//number of bonds
 var $atom;//molecule structure
 var $prop;//property structure
 var $dist;//distances structure
 var $e;//seed
 var $f;//forcefield
 var $m;//molecule number
 var $p;//paths structure
 var $s;//sys
 var $v;//view
 function m_c(){
  $this->a=0;$this->b=0;$this->p[0][0][0]=0;//atomul 0, calea 0, varful 0 e 0
 }
}
class fragments{
 var $a;//number of atoms
 var $f;//fragments structure
 var $d;//substructura de distanta
 function fragments(&$mol){//!!!!!!!!!!!!!de implementat cu foreach
  $this->a=$mol->a;
  foreach ($mol->atom as $atom_i){//fiecare varf i
   foreach ($mol->atom as $atom_j){//fiecare varf j
    $this->f[0][$atom_i[0]][$atom_j[0]][0]=$atom_i[0];
   }//primul varf - implicit
   $this->f[0][$atom_i[0]][$atom_i[0]][0]=0;
  }//genereaza fragmentele minimale - ok
 }
}
class models{
 var $m_t; var $m_d; var $m_s; var $fty; var $pty; var $dty; var $smo; var $lty;
 function models(&$poo){
  $this->m_t = array("mMDP","CHMEGQ","DdOoPpQqJjKkLlVEWwFfSsTt","RrMmDd");
  $this->m_d[0]="pr2m_".strpos($this->m_t[2],$poo["Df"]);
  $this->m_s[0]="prf_dmr_".strpos($this->m_t[3],$poo["Im"]);
  $this->fty=strpos($this->m_t[0],$poo["Fc"]);
  $this->pty=strpos($this->m_t[1],$poo["Ap"]);
  $this->dty=strpos("gt",$poo["Do"]);
```

```
  $this->smo=strpos("mMmNSPsAaBbGgFfHhIi",$poo["Sf"]);
  $this->lty=strpos("IiAaLl",$poo["Lo"]);
  $this->m_t = array($poo["Fc"],$poo["Ap"],$poo["Df"],$poo["Im"]);
 }
}
$m = new m_c;
read_mol_new($m,$file_data,"z_atom_data.txt");
tree_mol($m,$tv);
$f = new fragments($m);
gen_f_max($f,$m);
gen_f_sdi($f,$m);
gen_f_fdi($f,$m,$tv);
unset($f->d);
echo("<b>Predictor's equation:</b><UL><LI>".$_POST["equation"]."</UL>");
echo("<b>MDF Members:</b><UL>");
$val_sum=$coef[0];
for($i=1;$i<count($name);$i++){
      $post["Do"] = $name[$i][6];
      $post["Ap"] = $name[$i][5];
      $post["Df"] = $name[$i][4];
      $post["Im"] = $name[$i][3];
      $post["Fc"] = $name[$i][2];
      $post["Sf"] = $name[$i][1];
      $post["Lo"] = $name[$i][0];
      $model = new models($post);
      $afisa=af_f_pf($f,$m,$model);
      echo("<LI>".$afisa);
      list($tmp,$val)=explode("=",$afisa);
      $val_sum += $coef[$i]*$val;
}
echo("</UL>");
echo("<b>Predicted activity:</b><UL><LI>");
echo($val_sum."</UL>");
unset($m);
unset($tv);
unset($f);
?>
```

Programul de mai sus folosește pentru predicție nucleul de bază al familiei MDF (*mdf_functions.php*) și încă o aplicație (*mdf_predictus.php*), a cărei sursă este redată mai jos:

```
<?
function af_f_pf(&$frag,&$mol,&$md){
 $contor=0;
 $tip_f=$md->fty;//tip de fragment (4)
 $tip_p=$md->pty;//tip de proprietate (6)
 $md_d_c=$md->m_d[0];//tip de descriptor (24)
 $md_i_c=$md->m_s[0];//tip de interactiune (6)
 $f_cnt=0;
 foreach ($mol->atom as $atom_i){//fiecare varf i
  $i=$atom_i[0];
  foreach ($mol->atom as $atom_j){//fiecare varf j
   $varf=$atom_j[0];
   if($i==$varf) continue;
   if(!is_array($frag->f[$tip_f][$i][$varf][0]))
    $fragment[0]=$frag->f[$tip_f][$i][$varf];
   else
    $fragment=$frag->f[$tip_f][$i][$varf];
    for($fr_i=0;$fr_i<count($fragment);$fr_i++){
     $tmpp = $md_i_c($tip_p,$md_d_c,$varf,$fragment[$fr_i],$mol);
     $p_array[0][$f_cnt]=$tmpp[0];//gt
     $p_array[1][$f_cnt]=$tmpp[1];
```

```
     $f_cnt++;
    }
   }
  }
 $afisa="";//$afisa="Fragments: ".$f_cnt."\r\n"; $afisa.="Fragmental properties:
".print_r($p_array[$md->dty],TRUE);
 $na_fprop="";
 $na_fprop=$md->m_t[0].$md->m_t[3].$md->m_t[2].$md->m_t[1];
 $afisa.=af_cumul($contor,$mol->a,$mol->b,$p_array,$na_fprop,$md->dty,$md-
>smo,$md->lty);
 unset($p_array);
 unset($frag);
 unset($mol);
 return $afisa;
}//genereaza si afiseaza proprietatile fragmentale
function
af_cumul(&$contor,$nr_at,$nr_bo,&$p_array,&$na_fprop,$dtydty,$smo,$lty){
 $metric="gt";
 $nr_frags=count($p_array[0]);
 for($tip_x=$dtydty;$tip_x<$dtydty+1;$tip_x++){
  $sum=0;
  $prd=1;
  $isu=0;
  $max=-1e100;
  $max_a=0;
  $min=1e100;
  $min_a=1e100;
  $nr_fprop=0;
  for($tip_y=0;$tip_y<$nr_frags;$tip_y++){
   if(!$p_array[$tip_x][$tip_y]) continue;
   if(!is_finite($p_array[$tip_x][$tip_y]))continue;
   $nr_fprop++;
   $sum += $p_array[$tip_x][$tip_y];
   $prd *= $p_array[$tip_x][$tip_y];
   $isu += 1/$p_array[$tip_x][$tip_y];
   if($p_array[$tip_x][$tip_y]>$max) $max=$p_array[$tip_x][$tip_y];
   if(abs($p_array[$tip_x][$tip_y])>$max_a)
$max_a=abs($p_array[$tip_x][$tip_y]);
   if($p_array[$tip_x][$tip_y]<$min) $min=$p_array[$tip_x][$tip_y];
   if(abs($p_array[$tip_x][$tip_y])<$min_a)
$min_a=abs($p_array[$tip_x][$tip_y]);
  }
  if($max==-1e100) $max=(float)"INF";
  $v[0]=$max;//max property
  $af[0]="m".$na_fprop.$metric[$tip_x];
  $v[1]=$max_a;//max_a property
  $af[1]="M".$na_fprop.$metric[$tip_x];
  if($min==1e100) $min=(float)"INF";
  $v[2]=$min;//min property
  $af[2]="n".$na_fprop.$metric[$tip_x];
  if($min_a==1e100) $min_a=(float)"INF";
  $v[3]=$min_a;//min_a property
  $af[3]="N".$na_fprop.$metric[$tip_x];
  $v[4]=$sum;//sum property
  $af[4]="S".$na_fprop.$metric[$tip_x];
  $v[5]=$prd;//product property
  $af[5]="P".$na_fprop.$metric[$tip_x];
  if($isu)
    $v[6]=1/$isu;//harmonic sum property
   else
    $v[6]=(float)"INF";
  $af[6]="s".$na_fprop.$metric[$tip_x];
  if($nr_fprop)
   $v[7]=$sum/$nr_fprop;//average mean property for valid fragments
```

```
  else
   $v[7]=(float)"INF";
  $af[7]="A".$na_fprop.$metric[$tip_x];
  $v[8]=$sum/$nr_frags;//average mean property for all fragments
  $af[8]="a".$na_fprop.$metric[$tip_x];
  $v[9]=$sum/$nr_at;//average mean property for all fragments
  $af[9]="B".$na_fprop.$metric[$tip_x];
  $v[10]=$sum/$nr_bo;//average mean property for all fragments
  $af[10]="b".$na_fprop.$metric[$tip_x];
  if($nr_fprop)
   $v[11]=pow(abs($prd),1/$nr_fprop);//geometric   mean   property   for   valid
fragments - without sign
  else
   $v[11]=1;
  $af[11]="G".$na_fprop.$metric[$tip_x];
  $v[12]=pow(abs($prd),1/$nr_frags);//geometric   mean   property   fragments   -
without sign
  $af[12]="g".$na_fprop.$metric[$tip_x];
  $v[13]=pow(abs($prd),1/$nr_at);//geometric mean property fragments - without
sign
  $af[13]="F".$na_fprop.$metric[$tip_x];
  $v[14]=pow(abs($prd),1/$nr_bo);//geometric mean property fragments - without
sign
  $af[14]="f".$na_fprop.$metric[$tip_x];
  if($isu)
   $v[15]=$nr_fprop/$isu;//harmonic mean property for valid fragments
  else
   $v[15]=(float)"INF";
  $af[15]="H".$na_fprop.$metric[$tip_x];
  if($isu)
   $v[16]=$nr_frags/$isu;//harmonic mean property for valid fragments
  else
   $v[16]=(float)"INF";
  $af[16]="h".$na_fprop.$metric[$tip_x];
  if($isu)
   $v[17]=$nr_at/$isu;//harmonic mean property for valid fragments
  else
   $v[17]=(float)"INF";
  $af[17]="I".$na_fprop.$metric[$tip_x];
  if($isu)
   $v[18]=$nr_bo/$isu;//harmonic mean property for valid fragments
  else
   $v[18]=(float)"INF";
  $af[18]="i".$na_fprop.$metric[$tip_x];
 }
 $ret="";//$ret = $af[$smo]." = ".$v[$smo]."\r\n";
 if($lty==0){
  $ret .= "I".$af[$smo]." = ".$v[$smo]."\r\n";
 }
 if($lty==1){
  $ret .= "i".$af[$smo]." = ".(1/$v[$smo])."\r\n";
 }
 if($lty==2){
  $ret .= "A".$af[$smo]." = ".(abs($v[$smo]))."\r\n";
 }
 if($lty==3){
  $ret .= "a".$af[$smo]." = ".(abs(1/$v[$smo]))."\r\n";
 }
 if($lty==4){
  $ret .= "L".$af[$smo]." = ".(log(abs($v[$smo])))."\r\n";
 }
 if($lty==5){
  $ret .= "l".$af[$smo]." = ".(log($v[$smo]))."\r\n";
 }
```

```
 return $ret;
}
?>
```

Leave-one-out analyst

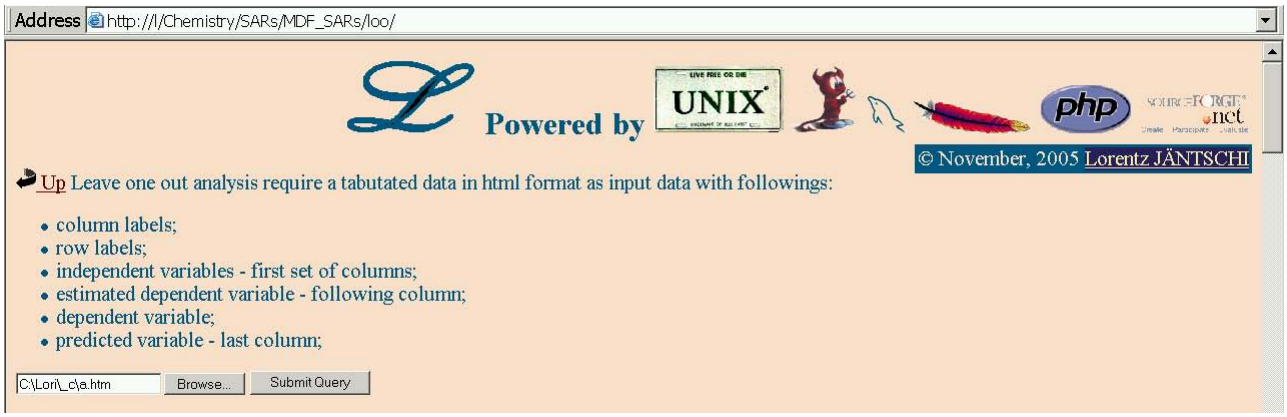
Aplicația leave-one-out este o aplicație expert pentru realizarea de experimente de predicție. Metodologia leave-one-out procedează astfel:

- se exclude o moleculă din set;
- cu moleculele rămase se construieşte modelul structură-activitate;
- cu ajutorul modelului structură-activitate se prezice activitatea pentru molecula exclusă;
- se repetă procedura de excludere/modelare/predicție pentru fiecare moleculă a setului;
- se reţin valorile prezise; se corelează valorile prezise cu valorile măsurate;
- scorul de corelație obținut se numeşte scorul leave-one-out.

Aplicația necesită ca date de intrare un tabel cu un anumit format, şi anume:

- liniile şi coloanele tabelului trebuie să aibă cap (cap de linie şi cap de coloană);
- prima coloană trebuie să conțină denumiri de compuşi;
- următoarele coloane trebuie să conțină descriptori moleculari;
- următoarea coloană trebuie să conțină variabila estimator;
- următoarea coloană trebuie să conțină variabila dependentă (măsurată);
- ultima coloană trebuie să conțină variabila predictor.

Programul produce analiza statistică a procedurii leave-one-out. Folosind datele de model din aplicație (figura următoare):



rezultatul execuției este următorul:

MOL	ISMMEQT	ISMMWHG	LADMKQT	INPRJQG	Y_EST	Y	Y_PRE
T1	2.0396E-2	7.0397E-4	1.4830	1.6344E-1	3.8295	3.82	3.8619
T2	1.9849E-2	5.212E-4	-1.5099	3.1038E-2	5.2003	5.2	5.2009
T3	1.7683E-2	4.2792E-4	1.8269	1.5977E-2	5.3298	5.34	5.3267
T4	1.5673E-2	3.2419E-4	2.3702	1.8778E-2	5.8340	5.83	5.8348
T5	1.9487E-2	4.0228E-4	1.0849	1.3049E-2	5.9636	6.01	5.9436

T6	1.7421E-2	3.2143E-4	7.3369E-1	5.5597E-2	6.4279	6.39	6.4385
T7	1.5468E-2	2.3758E-4	2.5927	8.8979E-2	6.7585	6.75	6.7610
T8	1.5478E-2	2.3847E-4	1.7558	5.8289E-2	6.7240	6.76	6.7183
T9	1.5474E-2	2.081E-4	3.7333	1.7769E-2	6.7403	6.74	6.7406
T10	1.3832E-2	1.7019E-4	2.7135	9.7269E-4	6.7797	6.76	6.7814
T11	1.3698E-2	1.6401E-4	2.1957	3.8135E-3	6.8536	6.78	6.8607
T12	1.4744E-2	1.6047E-4	4.2013	4.5768E-2	7.0711	7.12	7.0611
T13	1.5685E-2	2.2481E-4	2.0715	3.6145E-2	6.8024	6.82	6.8006
T14	1.4782E-2	1.8633E-4	3.2787	3.081E-2	6.8685	6.74	6.8787
T15	1.3289E-2	1.543E-4	2.1752	2.657E-3	6.8579	6.89	6.8544
T16	1.4048E-2	1.6089E-4	3.9860	4.565E-3	6.8263	6.95	6.8074
T17	1.3745E-2	1.5536E-4	1.6612	1.7316E-2	7.0175	7.01	7.0190
T18	1.3317E-2	1.4414E-4	3.3191	9.1285E-3	6.8923	6.87	6.8939
T19	1.25E-2	1.2782E-4	2.7222	2.1318E-2	6.9542	6.97	6.9526
T20	1.3427E-2	1.4622E-4	4.3633	5.5724E-2	6.9547	6.94	6.9563
T21	1.2885E-2	1.3033E-4	4.6685	9.5318E-2	7.0818	7.21	7.0469
T22	1.2405E-2	1.1766E-4	4.9193	7.7976E-2	7.0330	7.01	7.0374
T23	1.1977E-2	1.065E-4	5.1262	5.0319E-2	6.9546	6.81	6.9757
T24	1.3987E-2	1.8486E-4	4.0294	2.7798E-2	6.6618	6.45	6.6811
T25	1.3279E-2	1.7069E-4	3.4599	2.0853E-2	6.6692	6.75	6.6642
T26	1.2897E-2	1.5468E-4	2.5351	5.4539E-3	6.7591	6.75	6.7603
T27	1.2662E-2	1.5654E-4	4.0967	1.909E-2	6.6238	6.71	6.6129
T28	1.2303E-2	1.4043E-4	3.3219	2.3987E-2	6.7665	6.88	6.7548
T29	1.1586E-2	1.3456E-4	2.7653	3.8801E-2	6.7588	6.7	6.7764
T30	1.2137E-2	1.524E-4	4.4315	5.9321E-2	6.6481	6.69	6.6422

```
Df = 25
Var = 4
Estimated:
SSe = 0.16240756
QSSe = 0.0805996426791087
r2est = 0.988502203409943
Fest = 537.326527471997
p_est = <1e-17
Predicted:
SSp = 0.21236335
QSSp = 0.0921657962586988
r2pre = 0.984983245490457
Fpre = 409.45702995597
p_pre = <1e-17
```

Rezultatele obținute au semificațiile după cum urmează:

- Df - numărul de grade de libertate (25 = 30 - 4 - 1);
- Var - numărul de variabile independente (iSMmEQt, iSMMWHg, LADmkQt, INPRJQg);
- SSe și SSp - suma pătratelor erorilor în estimare și respectiv predicție;
- QSSe și QSSp - suma pătratelor erorilor din regresia de estimare și respectiv predicție;
- r2est și r2pre - coeficientul de determinare (pătratul coeficientului de corelație) în estimare și respectiv predicție;

- Fest şi Fpre - parametrul Fisher asociat sumelor de erori din estimare şi respectiv predicție;
- p_est şi p_pre - probabilitatea asociată valorii parametrului Fisher din distribuția F pentru estimare şi respectiv predicție.

Sursa aplicației este următoarea:

```
<?
$definition_page["up_dir"]="";
$definition_page["auto_index"]=FALSE;
$definition_page["copyright"]=array("November, 2005","lori");
include("/usr/home/www/data/definitions.php");
include("functions.php");
if(!isset($_FILES)){
	$a=strip_tags(strtoupper(file_get_contents("mysql_query_php_Triazines.htm"
)),"<TR><TD>");
	echo("File: mysql_query_php_Triazines.htm<br>\r\n");
}else{
	$a=strip_tags(strtoupper(file_get_contents($_FILES['file']['tmp_name'])),"
<TR><TD>");
	echo("File: ".$_FILES['file']['name']."<br>\r\n");
}
$b=explode("<TR>",$a);
echo("<table border=1>");
for($i=1;$i<count($b);$i++){
	$b[$i]=explode("<TD>",$b[$i]);
}
for($i=1;$i<count($b);$i++){
	for($j=1;$j<count($b[$i]);$j++){
		$b[$i][$j]=trim($b[$i][$j]);
	}
}
for($i=1;$i<count($b);$i++){
	echo("\r\n<tr>");
	for($j=1;$j<count($b[$i]);$j++){
		echo("<td>".$b[$i][$j]);
	}
}
echo("\r\n</table>");
$df=count($b)-count($b[1])+2;
echo("Df = ".$df."<BR>\r\n");
$var=count($b[1])-5;
echo("Var = ".$var."<BR>\r\n");
$ssp=0;$sse=0;
for($i=1;$i<count($b);$i++){
	$ssp+=pow($b[$i][count($b[1])-2]-$b[$i][count($b[1])-1],2);
	$sse+=pow($b[$i][count($b[1])-2]-$b[$i][count($b[1])-3],2);
	$y_mas[] = $b[$i][count($b[1])-2];
	$y_est[] = $b[$i][count($b[1])-3];
	$y_pre[] = $b[$i][count($b[1])-1];
}
array_shift($y_mas);array_shift($y_est);array_shift($y_pre);
echo("<b>Estimated:</b><br>");
echo("SSe = ".$sse."<BR>\r\n");
$sse /= $df;
$sse = pow($sse,0.5);
echo("QSSe = ".$sse."<BR>\r\n");
echo("r2est = ".coef_r($y_mas,$y_est)."<BR>\r\n");
$cFe = coef_F($y_mas,$y_est,$var);
echo("Fest = ".$cFe."<BR>\r\n");
$pest=coef_pF($var,$df+$var,$cFe);
if($pest==0) $pest="<1e-17";
echo("p_est = ".$pest."<BR>\r\n");
```

```
echo("<b>Predicted:</b><br>");
echo("SSp = ".$ssp."<BR>\r\n");
$ssp /= $df;
$ssp = pow($ssp,0.5);
echo("QSSp = ".$ssp."<BR>\r\n");
echo("r2pre = ".coef_r($y_mas,$y_pre)."<BR>\r\n");
$cFp = coef_F($y_mas,$y_pre,$var);
echo("Fpre = ".$cFp."<BR>\r\n");
$ppre=coef_pF($var,$df+$var,$cFp);
if($ppre==0) $ppre="<1e-17";
echo("p_pre = ".$ppre."<BR>\r\n");
?>
```

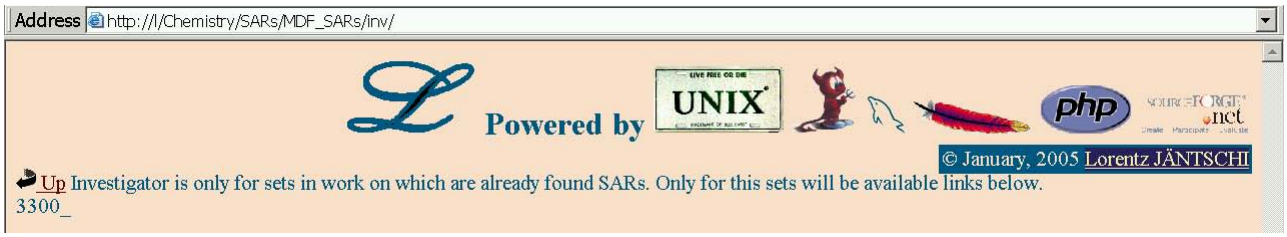
Aplicația de mai sus folosește o librărie de funcții (*functions.php*) a cărei sursă este următoarea:

```
<?
include("statistics/FDistribution.php");
function m1(&$v){
 $rez=0;
 $n=count($v);
 for($i=0;$i<$n;$i++)
  $rez+=$v[$i];
 return $rez/$n;
}
function m2(&$v,&$u){
 $rez=0;
 $n=count($v);
 for($i=0;$i<$n;$i++)
  $rez+=$v[$i]*$u[$i];
 return $rez/$n;
}
function coef_r(&$y1,&$y2){
  $my1=m1($y1);
  $dy2=m2($y1,$y1)-$my1*$my1;
  $mx1=m1($y2);
  $mxy=m2($y2,$y1);
  $m2x=$mx1*$mx1;
  $mx2=m2($y2,$y2);
  $dx2=$mx2-$m2x;
  $r2=pow($mxy-$mx1*$my1,2)/($dx2*$dy2);
  return $r2;
}
function coef_F(&$y1,&$y2,$df_r){
  $my1=m1($y1);
  $T_SS=0;
  for($i=0;$i<count($y1);$i++)
   $T_SS+=pow($y1[$i]-$my1,2);
  $E_SS=0;
  for($i=0;$i<count($y1);$i++)
   $E_SS+=pow($y1[$i]-$y2[$i],2);
  $R_SS=$T_SS-$E_SS;
  $T_df=count($y1)-1;
  $R_df=$df_r;
  $E_df=$T_df-$R_df;
  if(!$R_df)die();
  if(!$E_SS)die();
  $F=$R_SS*$E_df/$R_df/$E_SS;
  return $F;
}
function coef_pF($df_r,$df_t,$F){
 $Fd = new FDistribution($df_r,$df_t-$df_r);
 return 1.0 - $Fd->CDF($F);
```

```
}
?>
```

MDF Investigator

Aplicația MDF Investigator este o aplicație dedicată lucrului cu seturi de compuşi. Ea operează pe seturi pe care s-au găsit relații structură-activitate şi seturi care sunt încă în lucru (se află în baza de date MDFSARtmp). Captura ecranului acestei aplicații este:



Sursa aplicației este redată mai jos:

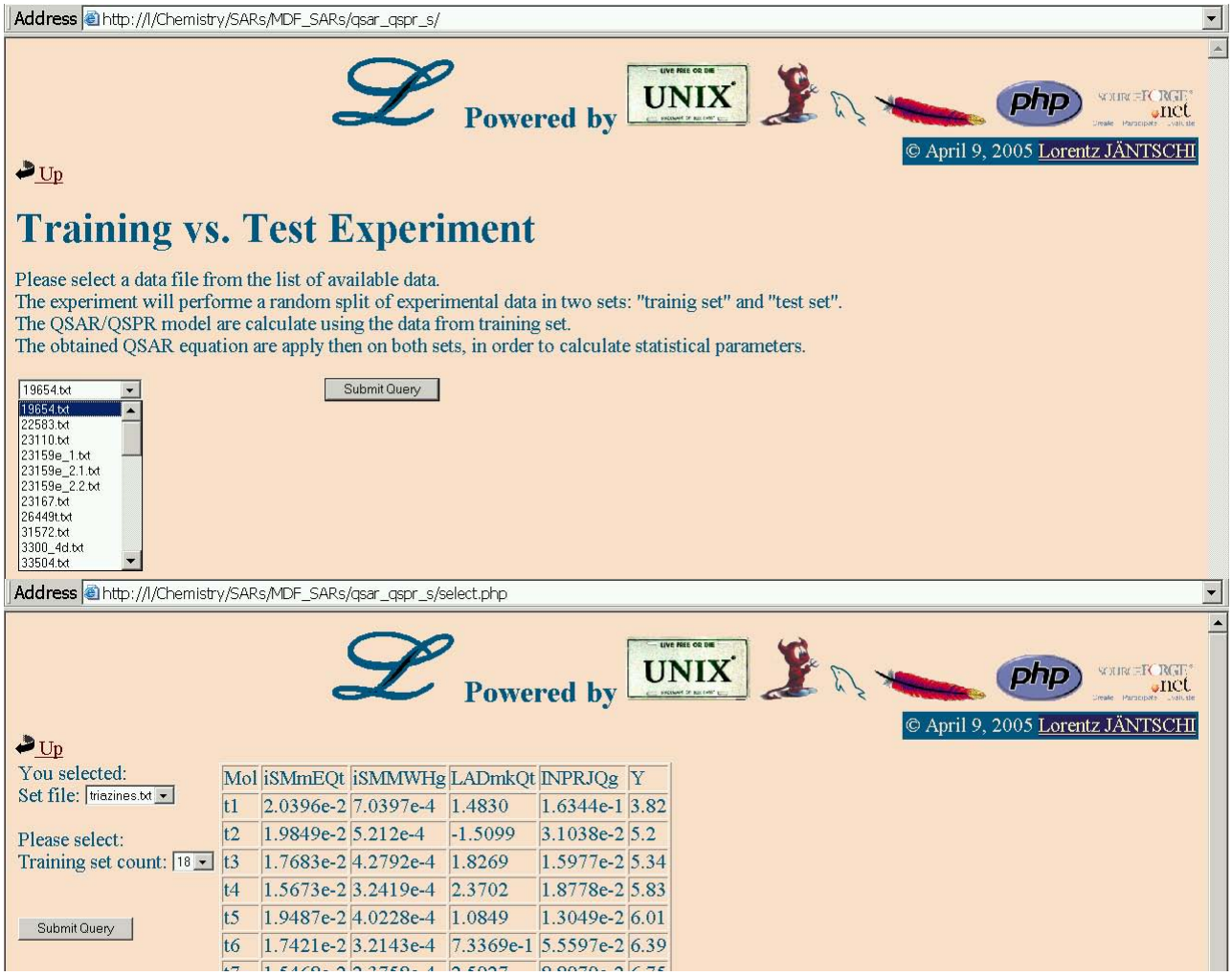
```
<?
$definition_page["auto_index"]=FALSE;
$definition_page["copyright"]=array("January, 2005","lori");
include("/usr/home/www/data/definitions.php");
if(array_key_exists("data",$_POST)){
      $_GET["t"]=$_POST["data"];
      $c=@mysql_connect($_POST["host"],$_POST["user"],$_POST["password"]);
      if(!$c){
            echo("mysql server not available.");
      }else{
            $q=mysql_query("USE `correlations`");
            if(!$q){
                  echo("mysql database not available.");
            }else if(array_key_exists("field",$_POST)){
                  for($i=0;$i<count($_POST["field"]);$i++){
                        $query="DELETE FROM `molecular_topology` WHERE `id` =
'".$_POST["field"][$i]."'";
                        $q=mysql_query($query);
                        if($q){
                              echo($query."<BR>\r\n");
                        }else{
                              echo("Query DELETE Error: ".$query."<BR>\r\n");
                        }
                  }
            }
            $q=mysql_query("OPTIMIZE TABLE `molecular_topology`");
            mysql_close($c);
      }
}
if(!array_key_exists("t",$_GET)){
      echo("Investigator is only for sets in work on which are already found
SARs. Only for this sets will be available links below.<br>");
 include("../0_mdf_definitions.php");
 $q=mysql_query("USE `".server_db_in_work."`");
 if(!$q)die("mysql database not available.");
 $q=mysql_query("SELECT DISTINCT `name` FROM `".server_table_ress."`");
 while($r=mysql_fetch_row($q)){
      $x[]=$r[0];
 }
 if(!isset($x))die("No data sets in temporary database.");
 for($i=0;$i<count($x);$i++){
      $y[$i]=0;
      $q1=mysql_query("SELECT * FROM `".$x[$i].server_table_tmpx."` LIMIT 1");
```

```
      if(!$q1){ $y[$i]=1; continue; }
      $q1=mysql_query("SELECT * FROM `".$x[$i].server_table_data."` LIMIT 1");
      if(!$q1){ $y[$i]=1; continue; }
      $q1=mysql_query("SELECT * FROM `".$x[$i].server_table_xval."` LIMIT 1");
      if(!$q1){ $y[$i]=1; continue; }
      $q1=mysql_query("SELECT * FROM `".$x[$i].server_table_yval."` LIMIT 1");
      if(!$q1){ $y[$i]=1; continue; }
 }
 for($i=0;$i<count($x);$i++){
      if($y[$i]) echo($x[$i]."<BR>\r\n");
      else echo("<A
HRef='".$_SERVER["PHP_SELF"]."?t=".$x[$i]."'>".$x[$i]."</A><BR>\r\n");
 }
 mysql_close();
 die();
}
$a=$_SERVER["HTTP_REFERER"];
$b=explode("/",$a);
unset($b[count($b)-1]);
$a=implode("/",$b)."/";
include("../0_mdf_definitions.php");
$a.="../i_mdf_query.php?database=".server_db_in_work."&set=".$_GET["t"];
$a=file_get_contents($a);
$b=explode("<HR size='30' color='black' bgcolor='black'>",$a);
echo($b[0]);
echo("<HR size='30' color='blue' bgcolor='blue'>");
$b[1]=strip_tags($b[1],"<tr><td><sub><sup>");
$a=explode("<tr>",$b[1]);
unset($b);
for($i=0;$i<count($a);$i++) if($a[$i]){
      $b[]=explode("<td>",$a[$i]);
}
$j=0;
for($i=0;$i<count($b);$i++){
      if(is_numeric($b[$i][2])){
            if($b[$i][2]){//".$j."
             $b[$i][0]="<input type='checkbox' name='field[]'
value='".$b[$i][2]."'>";
             $j++;
         }else{
             $b[$i][0]=$b[$i][2];
         }
      }else{
           $b[$i][0]="X?";
      }
}
unset($a);
for($i=0;$i<count($b);$i++){
      $a[$i]="<td>".implode("<td>",$b[$i]);
}
unset($b);
$b="<table border=1>\r\n<tr>".implode("\r\n<tr>",$a)."\r\n</table>\r\n";
echo("<form method='post' action='".$_SERVER["PHP_SELF"]."'>\r\n");
echo($b);
echo("data:<input type='text' name='data' value='".$_GET["t"]."'><BR>\r\n");
echo("host:<input type='text' name='host'
value='".server_ip_address."'><BR>\r\n");
echo("user:<input type='text' name='user' value='read'><BR>\r\n");
echo("pass:<input type='password' name='password' value=''><BR>\r\n");
echo("<input type='submit'>\r\n</form>");
?>
```

Traning vs. Test Experiment

Aplicația *Traning vs. Test Experiment* este o aplicație expert specializată pe realizarea de experimente de comportare a modelului SAR obținut (stabilitate, capacitate de predicție, calitatea regresiei, ș.a.m.d.). Aplicația permite ruperea setului investigat în două subseturi: setul școală și setul test; setul școală este folosit pentru a învăța sistemul (obține ecuația SAR); setul test este folosit pentru a testa sistemul (aplică ecuația SAR).

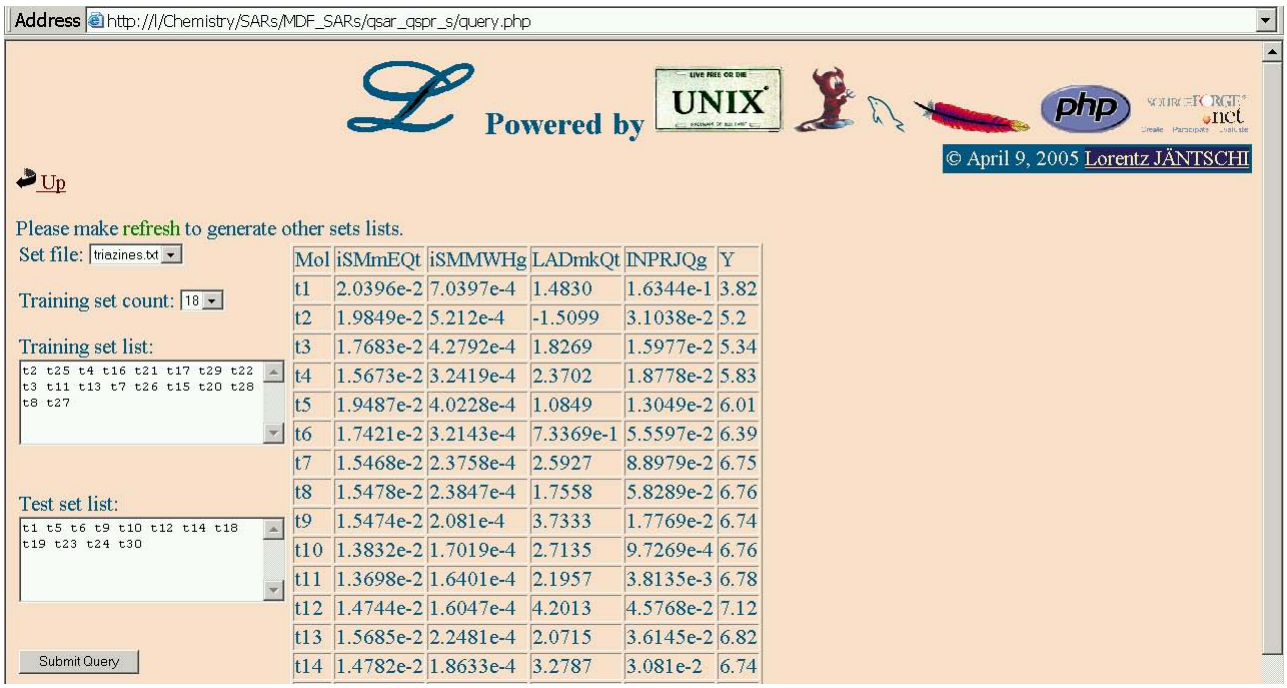
Pagina de start a aplicației este redată mai jos. Alegând ultima opțiune din listă, aplicația produce:



Aplicația permite alegerea unui număr variabil de molecule în setul de învățare; alegând 18 la acest pas, obținem o extragere randomizată a moleculelor în set test și set școală (figura următoare).

O nouă împărțire a setului ales în subset test și subset școală se poate obține fie mergând înapoi și apăsând din nou butonul *Submit Query* fie apăsând butonul *Refresh* de pe navigatorul de pagină.

Butonul următor *Submit Query* folosește moleculele din lista de învățare pentru obținerea modelului SAR (coeficienții modelului) și aplică modelul SAR pe setul test.



Mol	iSMmEQt	iSMMWHg	LADmkQt	INPRJQg	Y
t1	2.0396e-2	7.0397e-4	1.4830	1.6344e-1	3.82
t2	1.9849e-2	5.212e-4	-1.5099	3.1038e-2	5.2
t3	1.7683e-2	4.2792e-4	1.8269	1.5977e-2	5.34
t4	1.5673e-2	3.2419e-4	2.3702	1.8778e-2	5.83
t5	1.9487e-2	4.0228e-4	1.0849	1.3049e-2	6.01
t6	1.7421e-2	3.2143e-4	7.3369e-1	5.5597e-2	6.39
t7	1.5468e-2	2.3758e-4	2.5927	8.8979e-2	6.75
t8	1.5478e-2	2.3847e-4	1.7558	5.8289e-2	6.76
t9	1.5474e-2	2.081e-4	3.7333	1.7769e-2	6.74
t10	1.3832e-2	1.7019e-4	2.7135	9.7269e-4	6.76
t11	1.3698e-2	1.6401e-4	2.1957	3.8135e-3	6.78
t12	1.4744e-2	1.6047e-4	4.2013	4.5768e-2	7.12
t13	1.5685e-2	2.2481e-4	2.0715	3.6145e-2	6.82
t14	1.4782e-2	1.8633e-4	3.2787	3.081e-2	6.74

Pentru seturile alese, se realizează analiza statistică a rezultatelor obținute (figura următoare):

Set file: triazines.txt
Training set count: 18
Training set: t2 t25 t4 t16 t21 t17 t29 t22 t3 t11 t13 t7 t26 t15 t20 t28 t8 t27
Test set: t1 t5 t6 t9 t10 t12 t14 t18 t19 t23 t24 t30
Training set data:

Mol	iSMmEQt	iSMMWHg	LADmkQt	INPRJQg	Y
t2	1.9849e-2	5.212e-4	-1.5099	3.1038e-2	5.2
t25	1.3279e-2	1.7069e-4	3.4599	2.0853e-2	6.75
t4	1.5673e-2	3.2419e-4	2.3702	1.8778e-2	5.83
t16	1.4048e-2	1.6089e-4	3.9860	4.565e-3	6.95
t21	1.2885e-2	1.3033e-4	4.6685	9.5318e-2	7.21
t17	1.3745e-2	1.5536e-4	1.6612	1.7316e-2	7.01
t29	1.1586e-2	1.3456e-4	2.7653	3.8801e-2	6.7
t22	1.2405e-2	1.1766e-4	4.9193	7.7976e-2	7.01
t3	1.7683e-2	4.2792e-4	1.8269	1.5977e-2	5.34
t11	1.3698e-2	1.6401e-4	2.1957	3.8135e-3	6.78
t13	1.5685e-2	2.2481e-4	2.0715	3.6145e-2	6.82
t7	1.5468e-2	2.3758e-4	2.5927	8.8979e-2	6.75
t26	1.2897e-2	1.5468e-4	2.5351	5.4539e-3	6.75
t15	1.3289e-2	1.543e-4	2.1752	2.657e-3	6.89
t20	1.3427e-2	1.4622e-4	4.3633	5.5724e-2	6.94
t28	1.2303e-2	1.4043e-4	3.3219	2.3987e-2	6.88
t8	1.5478e-2	2.3847e-4	1.7558	5.8289e-2	6.76
t27	1.2662e-2	1.5654e-4	4.0967	1.909e-2	6.71

QSAR/QSPR: Y_EST = 5.4844+216.24*iSMmEQt+-9117.5*iSMMWHg+-0.0415*LADmkQt+2.4574*INPRJQg
Coefficient of determination r^2 = 0.9911
Fisher test value F = 361.92

Probability of wrong (from F) p_F = 0.00000 % (3.48832074337224E-13)
Test set data:

Mol	iSMmEQt	iSMMWHg	LADmkQt	INPRJQg	Y
t1	2.0396e-2	7.0397e-4	1.4830	1.6344e-1	3.82
t5	1.9487e-2	4.0228e-4	1.0849	1.3049e-2	6.01
t6	1.7421e-2	3.2143e-4	7.3369e-1	5.5597e-2	6.39
t9	1.5474e-2	2.081e-4	3.7333	1.7769e-2	6.74
t10	1.3832e-2	1.7019e-4	2.7135	9.7269e-4	6.76
t12	1.4744e-2	1.6047e-4	4.2013	4.5768e-2	7.12
t14	1.4782e-2	1.8633e-4	3.2787	3.081e-2	6.74
t18	1.3317e-2	1.4414e-4	3.3191	9.1285e-3	6.87
t19	1.25e-2	1.2782e-4	2.7222	2.1318e-2	6.97
t23	1.1977e-2	1.065e-4	5.1262	5.0319e-2	6.81
t24	1.3987e-2	1.8486e-4	4.0294	2.7798e-2	6.45
t30	1.2137e-2	1.524e-4	4.4315	5.9321e-2	6.69

Coefficient of determination r^2 = 0.9898
Fisher test value F = 87.400
Probability of wrong (from F) p_F = 0.00047 % (4.69603014918718E-06)

Sursele aplicațiilor sunt redate mai jos în următoarea ordine:

- index.php
- select.php
- query.php
- result.php

```
<?
$definition_page["auto_index"]=FALSE;
$definition_page["copyright"]=array("April 9, 2005","lori");
include("/usr/home/www/data/definitions.php");
?>
<h1>
Training vs. Test Experiment
</h1>
Please select a data file from the list of available data.
<BR>
The experiment will performe a random split of experimental data in two sets:
"trainig set" and "test set".
<BR>
The QSAR/QSPR model are calculate using the data from training set.
<BR>
The obtained QSAR equation are apply then on both sets, in order to calculate
statistical parameters.
<BR>
<?
$a=`ls -al *.txt`;
$b=explode("\n",$a);
?>
<form method='post' action='select.php'>
<table width='50%'>
<tr>
<td>
<select name='select'>
<?
for($i=0;$i<count($b);$i++){
```

```
 $c=explode(" ",$b[$i]);
 $d=$c[count($c)-1];
 if($d) echo("<option value='".$d."'>".$d);
}
echo("</select><td><input type='submit'></table></form>");
?>
<?
$definition_page["auto_index"]=FALSE;
$definition_page["copyright"]=array("April 9, 2005","lori");
include("/usr/home/www/data/definitions.php");
include("functions.php");
get_all_data("select",$_POST,$e);
?>
<table border=0>
<tr>
<td valign='top'>
<form method='post' action='query.php'>
You selected:
<BR>
Set file:
<select name='select'>
<option value='<? echo($_POST['select']); ?>' selected><?
echo($_POST['select']); ?>
</select>
<BR>
<BR>
Please select:
<BR>
Training set count:
<select name='test'>
<?
for($i=count($e[0])-1;$i<count($e);$i++){
 echo("<option value='".$i."'");
 if($i==(int)((count($e[0])+count($e))/2)) echo(" selected ");
 echo(">".$i);
}
?>
</select>
<BR>
<BR>
<input type='submit'>
</form>
<td>
<?
af($e,"");
echo("</table>");
?>
<?
$definition_page["auto_index"]=FALSE;
$definition_page["copyright"]=array("April 9, 2005","lori");
include("/usr/home/www/data/definitions.php");
include("functions.php");
get_all_data("select",$_POST,$e);
?>
<br><br>
Please make <font color='green'>refresh</font> to generate other sets lists.
<BR>
<table border=0>
<tr>
<td valign='top'>
<form method='post' action='result.php'>
Set file:
<select name='select'>
```

```
<option value='<? echo($_POST['select']); ?>' selected><?
echo($_POST['select']); ?>
</select>
<BR>
<BR>
Training set count:
<select name='test'>
<option value='<? echo($_POST['test']); ?>' selected><? echo($_POST['test']); ?>
</select>
<?
for($i=0;$i<$_POST['test'];$i++){
 do{
  $ok=0;
  $trs[$i]=rand(1,count($e)-1);
  for($j=0;$j<count($trs)-1;$j++)
   if($trs[$j]==$trs[$i]){
      $ok=1;
      break;
   }
 }while($ok);
}
for($i=1;$i<count($e);$i++)
 $f[$i]=$i;
for($i=0;$i<count($trs);$i++){
 unset($f[$trs[$i]]);
 $trs[$i]=$e[$trs[$i]][0];
}
$tr_s=implode(" ",$trs);
foreach($f as $kf => $vf)
 $f[$kf]=$e[$vf][0];
$te_s=implode(" ",$f);
?>
<BR>
<BR>
Training set list:
<BR>
<textarea name='training' rows='5' cols='30'>
<?
echo($tr_s);
?>
</textarea>
<BR>
<BR>
Test set list:
<BR>
<textarea name='rest' rows='5' cols='30'>
<?
echo($te_s);
?>
</textarea>
<BR>
<BR>
<input type='submit'>
</form>
<td>
<?
af($e,"");
echo("</table>");
?>
<?
$definition_page["auto_index"]=FALSE;
$definition_page["copyright"]=array("April 9, 2005","lori");
include("/usr/home/www/data/definitions.php");
include("functions.php");
```

```
include("mlr.php");
echo("<br><br>");
get_all_data("select",$_POST,$e);
echo("Set file: ".$_POST['select']."<BR>");
echo("Training set count: ".$_POST['test']."<BR>");
echo("Training set: ".$_POST['training']."<BR>");
echo("Test set: ".$_POST['rest']."<BR>");
$tra=explode(" ",$_POST['training']);
$tes=explode(" ",$_POST['rest']);
sub_test($tra,$e,$t_tra);
sub_test($tes,$e,$t_tes);
af($t_tra,"Training set data:");
echo("QSAR/QSPR: ");
for($i=1;$i<count($t_tra[0])-1;$i++){
 $t[$i-1]=$t_tra[0][$i];
 for($j=1;$j<count($t_tra);$j++)
  $x[$i-1][$j-1]=$t_tra[$j][$i];
}
for($j=1;$j<count($t_tra);$j++)
 $y[$j-1]=$t_tra[$j][count($t_tra[0])-1];
get_coefs($y,$x,$c);
for($j=0;$j<count($c);$j++)
 $c[$j]=spalare($c[$j]);//echo($c[$j]." ");
echo(put_coef_ind($c,$t,"EST")."<BR>");
calc_eq_1($c,$x,$y_pred);
echo("Coefficient of determination r<sup>2</sup> =
".sprintf("%.4f",coef_r($y,$y_pred))."<BR>");
$coef_F=coef_F($y,$y_pred,count($c)-1);
echo("Fisher test value F = ".spalare($coef_F)."<BR>");
$coef_pF=coef_pF(count($c)-1,count($y)-1,$coef_F);
echo("Probability of wrong (from F) p<sub>F</sub> =
".sprintf("%1.5f",$coef_pF*100)." % (".$coef_pF.")<BR>");
for($i=1;$i<count($t_tes[0])-1;$i++){
 $tt[$i-1]=$t_tes[0][$i];
 for($j=1;$j<count($t_tes);$j++)
  $xx[$i-1][$j-1]=$t_tes[$j][$i];
}
for($j=1;$j<count($t_tes);$j++)
 $yy[$j-1]=$t_tes[$j][count($t_tes[0])-1];
if(count($t_tes)>1){
 af($t_tes,"Test set data:");
 calc_eq_1($c,$xx,$yy_pred);
 echo("Coefficient of determination r<sup>2</sup> =
".sprintf("%.4f",coef_r($yy,$yy_pred))."<BR>");
 $coef_F=coef_F($yy,$yy_pred,count($c)-1);
 echo("Fisher test value F = ".spalare($coef_F)."<BR>");
 $coef_pF=coef_pF(count($c)-1,count($yy)-1,$coef_F);
 echo("Probability of wrong (from F) p<sub>F</sub> =
".sprintf("%1.5f",$coef_pF*100)." % (".$coef_pF.")<BR>");
}else echo("You selected all data in training set, thus no Test set data
here.<BR>");
?>
```

Aplicațiile de mai sus folosesc modulele *functions.php* şi *mlr.php* ale căror surse sunt redate mai jos (în această ordine):

```
<?
function get_all_data($text,&$tab,&$e){
 if(!array_key_exists($text,$tab)) die("no select");
 $a=file_get_contents($tab[$text]);
 $b=explode("\r\n",$a);
 for($j=0,$i=0;$i<count($b);$i++)
  if(!$b[$i]) continue; else
```

```
   $c[$j++]=explode(" ",$b[$i]);
 for($i=0;$i<count($c);$i++)
  for($k=0,$j=0;$j<count($c[$i]);$j++)
   if($c[$i][$j]==="") continue; else
    $d[$i][$k++]=$c[$i][$j];
 for($i=0;$i<count($d);$i++){
  for($j=0;$j<count($d[$i])-2;$j++)
   $e[$i][$j]=$d[$i][$j];
  $e[$i][count($d[$i])-3]=$d[$i][count($d[$i])-2];
 }
}
function af(&$e,$msg){
 if($msg) echo($msg."<BR>");
?>
<table border=1>
<?
for($i=0;$i<count($e);$i++){
 echo("<tr>");
 for($j=0;$j<count($e[$i]);$j++)
  echo("<td>".$e[$i][$j]);
}
?>
</table>
<?
}
?>
<?
include 'statistics/FDistribution.php';
function m1(&$v){
 $rez=0;
 $n=count($v);
 for($i=0;$i<$n;$i++)
  $rez+=$v[$i];
 return $rez/$n;
}
function m2(&$v,&$u){
 $rez=0;
 $n=count($v);
 for($i=0;$i<$n;$i++)
  $rez+=$v[$i]*$u[$i];
 return $rez/$n;
}
function get_coefs(&$y,&$x,&$b){
 $b[0] = m1($y);//x[ind][mol]
 $a[0][0] = 1.0;
 for($k=1;$k<=count($x);$k++){
  $a[0][$k]  = m1($x[$k-1]);
  $a[$k][0]  = $a[0][$k];
  $a[$k][$k] = m2($x[$k-1],$x[$k-1]);
  $b[$k] = m2($x[$k-1],$y);
  for($l=$k+1;$l<=count($x);$l++){
   $a[$k][$l] = m2($x[$k-1],$x[$l-1]);
   $a[$l][$k] = $a[$k][$l];
  }
 }
 $a[0][0] = gauss($b,$a);
}
function gauss(&$b,&$a){
 for($i=0;$i<count($a);$i++){
  $m=$i;//caut max in coloana
  for($j=$i+1;$j<count($a);$j++)
    if(abs($a[$i][$j])>abs($a[$i][$m]))$m=$j;
  if($a[$i][$m]==0)return -1;
  if($m != $i){//inlocuiesc linii
```

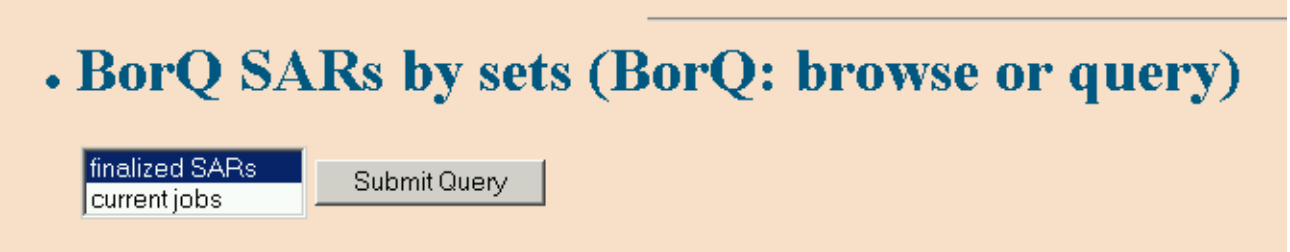
```
    $tmp=$b[$m];$b[$m]=$b[$i];$b[$i]=$tmp;
    for($j=0;$j<count($a);$j++){
      $tmp=$a[$m][$j];$a[$m][$j]=$a[$i][$j];$a[$i][$j]=$tmp;
    }
  }
  if(!abs($a[$i][$i])) return -1;
  for($j=$i+1;$j<count($a);$j++)//impart linie
    $a[$i][$j]/=$a[$i][$i];
  $b[$i]/=$a[$i][$i];
  $a[$i][$i]=1;
  for($j=$i+1;$j<count($a);$j++){//fac 0 sub diagonala
    $b[$j]-=$a[$j][$i]*$b[$i];
    for($k=count($a)-1;$k>$i;$k--)
      $a[$j][$k]-=$a[$j][$i]*$a[$i][$k];
  }
 }
 for($i=count($a)-1;$i>0;$i--)//fac 0 peste diagonala
  for($j=$i-1;$j>=0;$j--)
    $b[$j]-=$a[$j][$i]*$b[$i];
 return 1;
}
function calc_eq(&$coef,&$val,&$rez){//predictie pentru fiecare molecula
 for($i=0;$i<count($val[1]);$i++){//a cata molecula
  $rez[$i]=$coef[0];
  for($j=1;$j<count($coef);$j++){//al catelea indice si coeficient
   $rez[$i]+=$coef[$j]*$val[$j][$i];
  }
  $rez[$i]=spalare($rez[$i]);
 }
}
function calc_eq_1(&$coef,&$val,&$rez){//predictie pentru fiecare molecula
 for($i=0;$i<count($val[0]);$i++){//a cata molecula
  $rez[$i]=$coef[0];
  for($j=1;$j<count($coef);$j++){//al catelea indice si coeficient
   $rez[$i]+=$coef[$j]*$val[$j-1][$i];
  }
  $rez[$i]=spalare($rez[$i]);
 }
}

function new_y(&$y_old,&$x_old,&$y_new){//x[ind][mol]
  $cate=count($x_old[1]);
  $cati=count($x_old);
  for($i=0;$i<$cate;$i++){
   $rest[0]=exclude_i($i,$y_old,$y_new_i);
   for($j=1;$j<=$cati;$j++)
    $rest[$j-1]=exclude_i($i,$x_old[$j],$x_new_i[$j-1]);
   get_coefs($y_new_i,$x_new_i,$c_new);
   calc_1v($c_new,$rest,$y_new[$i]);//predictie pentru o molecula
  }
}
function calc_1v(&$coef,&$val,&$rez){//predictie pentru o molecula
 $rez=$coef[0];
 for($j=1;$j<count($coef);$j++){//al catelea indice si coeficient
  $rez+=$coef[$j]*$val[$j-1];
 }
 $rez=spalare($rez);
}
function exclude_i($i,&$data,&$new_data){
  for($j=0;$j<$i;$j++)
   $new_data[$j]=$data[$j];
  for($j=$i+1;$j<count($data);$j++)
   $new_data[$j-1]=$data[$j];
  return $data[$i];
```

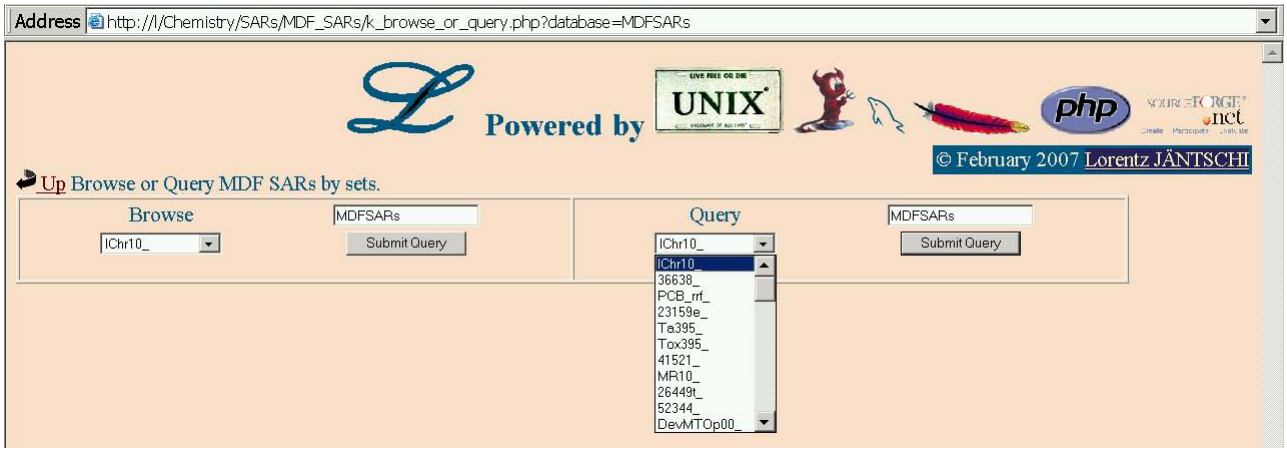
```
}
function spalare($value){
 if(!is_finite($value))
  return (float)"INF";
 $log_cifre=log(abs($value),10);
 if($log_cifre<0) $cifre=(int)$log_cifre;
 if($log_cifre>0) $cifre=(int)($log_cifre+1);
 $new2value=((int)($value*pow(10,5-$cifre)))*pow(10,-4);
 $new3value=$new2value."e".($cifre-1);
 if(($cifre>0)&&($cifre<6)) $new3value=sprintf("%.".(5-$cifre)."f",$new3value);
 if(($cifre<0)&&($cifre>-6)) $new3value=sprintf("%.".(5-
abs($cifre))."f",$new3value);
 return $new3value;
}
function sub_test(&$list,&$data,&$out){
 $out[0]=$data[0];
 for($i=0;$i<count($list);$i++)
  for($j=1;$j<count($data);$j++)
   if($list[$i]==$data[$j][0]){
      $out[$i+1]=$data[$j];
      break;
   }
}
function coef_r(&$y1,&$y2){
  $my1=m1($y1);
  $dy2=m2($y1,$y1)-$my1*$my1;
  $mx1=m1($y2);
  $mxy=m2($y2,$y1);
  $m2x=$mx1*$mx1;
  $mx2=m2($y2,$y2);
  $dx2=$mx2-$m2x;
  $r2=pow($mxy-$mx1*$my1,2)/($dx2*$dy2);
  return $r2." (n = ".count($y1).")";
}
function coef_F(&$y1,&$y2,$df_r){
  $my1=m1($y1);
  $T_SS=0;
  for($i=0;$i<count($y1);$i++)
   $T_SS+=pow($y1[$i]-$my1,2);
  $E_SS=0;
  for($i=0;$i<count($y1);$i++)
   $E_SS+=pow($y1[$i]-$y2[$i],2);
  $R_SS=$T_SS-$E_SS;
  $T_df=count($y1)-1;
  $R_df=$df_r;
  $E_df=$T_df-$R_df;
  if(!$R_df)die();
  if(!$E_SS)die();
  $F=$R_SS*$E_df/$R_df/$E_SS;
  return $F;
}
function coef_pF($df_r,$df_t,$F){
 $Fd = new FDistribution($df_r,$df_t-$df_r);
 return 1.0 - $Fd->CDF($F);
}
function put_coef_ind(&$coef,&$indx,$msg){
 $eq="Y_".$msg." = ".$coef[0];
 for($i=1;$i<count($coef);$i++){
  $eq.="+".$coef[$i]."*".$indx[$i-1];
 }
 return $eq;
}
?>
```

BorQ SARs

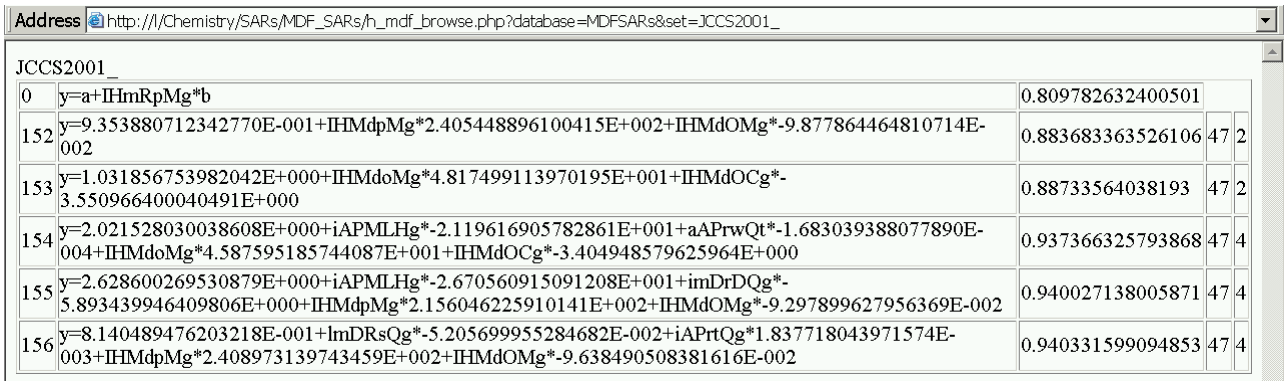
Aplicația *BorQ SARs* este specializată în interogarea bazei de date cu relații structură-activitate și este capabilă să facă această actinue atât asupra relațiilor stocate în baza de date permanentă (MDFSARs) cât și în baza de date temporară (MDFSARtmp), vezi figura următoare:



Alegând prima opțiune (figura de mai sus) și apoi ultima opțiune (figura de mai jos, întâi pentru browse și apoi pentru query) se obține prelucrarea informației din baza de date (figurile următoare).



Pentru browse:



0	y=a+IHmRpMg*b	0.809782632400501		
152	y=9.353880712342770E-001+IHMdpMg*2.405448896100415E+002+IHMdOMg*-9.877864464810714E-002	0.883683363526106	47	2
153	y=1.031856753982042E+000+IHMdoMg*4.817499113970195E+001+IHMdOCg*-3.550966400040491E+000	0.88733564038193	47	2
154	y=2.021528030038608E+000+iAPMLHg*-2.119616905782861E+001+aAPrwQt*-1.683039388077890E-004+IHMdoMg*4.587595185744087E+001+IHMdOCg*-3.404948579625964E+000	0.937366325793868	47	4
155	y=2.628600269530879E+000+iAPMLHg*-2.670560915091208E+001+imDrDQg*-5.893439946409806E+000+IHMdpMg*2.156046225910141E+002+IHMdOMg*-9.297899627956369E-002	0.940027138005871	47	4
156	y=8.140489476203218E-001+lmDRsQg*-5.205699955284682E-002+iAPrtQg*1.837718043971574E-003+IHMdpMg*2.408973139743459E+002+IHMdOMg*-9.638490508381616E-002	0.940331599094853	47	4

Pentru query:

JCCS2001_
MDF Size: 131328
dy2=0.392236758714357
r2(IHmRpMg) = 0.8098
dx2=3.09632858125781E-08
y = -16.511+3202.8*IHmRpMg

Mol	IHmRpMg	Y_est	Y	Y_pred
af	5.887e-3	2.3438	1.72	2.3631
al	5.701e-3	1.7481	1.7	1.7496

av	5.701e-3	1.7481	1.16	1.7636
ff	6.1305e-3	3.1237	3.1	3.1268
fg	5.8315e-3	2.1661	1.77	2.1758
fl	6.0246e-3	2.7845	2.87	2.7790
fp	5.9829e-3	2.6510	2.7	2.6486
fy	5.9531e-3	2.5555	3.13	2.5302
gf	5.8315e-3	2.1661	1.8	2.1750
gi	5.6236e-3	1.5002	1.7	1.4927
gl	5.6236e-3	1.5002	1.68	1.4934
gp	5.5358e-3	1.2190	1.35	1.2107
gv	5.5358e-3	1.2190	1.13	1.2250
gw	5.8864e-3	2.3419	1.89	2.3559
gy	5.6675e-3	1.6408	1.77	1.6371
ia	5.701e-3	1.7481	1.68	1.7501
i_d	5.4638e-3	9.8845e-1	1.37	9.5153e-1
ie	5.5293e-3	1.1982	1.37	1.1868
ig	5.6236e-3	1.5002	1.68	1.4934
ii	5.887e-3	2.3438	2.26	2.3466
ik	5.837e-3	2.1837	1.65	2.1969
il	5.887e-3	2.3438	2.26	2.3466
in	5.5708e-3	1.3311	1.49	1.3229
ip	5.8315e-3	2.1661	2.4	2.1607
iq	5.6327e-3	1.5294	1.49	1.5310
is	5.5319e-3	1.2065	1.49	1.1878
it	5.6675e-3	1.6408	1.49	1.6456
iv	5.8315e-3	2.1661	2.05	2.1691
iw	6.0466e-3	2.8550	3.05	2.8403
la	5.701e-3	1.7481	1.72	1.7491
lf	6.0246e-3	2.7845	2.75	2.7871
lg	5.6236e-3	1.5002	1.72	1.4919
ll	5.887e-3	2.3438	2.35	2.3439
lw	6.0466e-3	2.8550	3.4	2.8135
ly	5.8732e-3	2.2996	2.46	2.2953
pa	5.6236e-3	1.5002	1.32	1.5074
pf	5.9829e-3	2.6510	2.8	2.6433
pi	5.8315e-3	2.1661	2.33	2.1624
pl	5.8315e-3	2.1661	2.22	2.1650
py	5.8283e-3	2.1558	1.8	2.1645
sl	5.5319e-3	1.2065	1.49	1.1878
va	5.6236e-3	1.5002	1.16	1.5137
vg	5.5358e-3	1.2190	1.19	1.2211
vl	5.8315e-3	2.1661	2	2.1703
vv	5.7699e-3	1.9688	1.71	1.9746
ww	6.1442e-3	3.1676	3.6	3.1123
yl	5.8732e-3	2.2996	2.4	2.2970

r2(Y,Y_EST) = 0.8098
r2(Y,Y_PRED) = 0.7921

r2 = 0.9403
id = 156
Indices = 4
Molecules = 47
r2(lmDRsQg) = 0.0320183033266092; dx2=9.68914348180733
r2(iAPrtQg) = 0.0112973589986294; dx2=8164.59769929977
r2(IHMdpMg) = 0.739782021807653; dx2=0.000128347121604344
r2(IHMdOMg) = 0.662751528392356; dx2=495.687834122231
y = 8.140489476203218E-001+-5.2056e-2*lmDRsQg+1.8377e-3*iAPrtQg+240.89*IHMdpMg+-9.6384e-2*IHMdOMg

Mol	lmDRsQg	iAPrtQg	IHMdpMg	IHMdOMg	Y_est	Y	Y_pred
af	-5.7426e-1	-199.92	4.7487e-2	103.33	1.9563	1.72	1.9783
al	-1.5310	-1.5216e-3	3.5221e-2	78.283	1.8329	1.7	1.8437
av	6.9031	1.729e-4	3.6111e-2	79.717	1.4700	1.16	1.5378
ff	6.7924	3.4491e-7	6.3041e-2	130.03	3.1135	3.1	3.1167
fg	-3.4650	-168.73	4.6636e-2	103.97	1.8974	1.77	1.9066
fl	2.0647	-2.0377e-1	5.2769e-2	110.11	2.8048	2.87	2.7979
fp	5.9955e-1	95.403	5.6856e-2	121.93	2.9021	2.7	2.9427
fy	-5.9141e-1	1.1929e-1	6.6563e-2	142.77	3.1186	3.13	3.1164
gf	-2.0907	-172.28	4.5237e-2	100.37	1.8293	1.8	1.8311
gi	-2.6781	-98.822	3.3567e-2	75.788	1.5530	1.7	1.5433
gl	-2.6561	-99.093	3.3676e-2	76.133	1.5444	1.68	1.5357
gp	-1.9720	-67.712	3.2627e-2	77.719	1.1609	1.35	1.1446
gv	-2.7181	-82.632	3.0748e-2	70.898	1.3771	1.13	1.3957
gw	8.3539	2.2264e-2	5.321e-2	117.27	1.8940	1.89	1.8951
gy	-2.4551	-193.07	4.6355e-2	105.61	1.5743	1.77	1.5520
ia	-1.5114	-114.20	3.6478e-2	81.374	1.6268	1.68	1.6242
i_d	-1.4143	-2.2637	4.1731e-2	98.727	1.4203	1.37	1.4335
ie	4.039e-1	-93.589	4.381e-2	102.41	1.3037	1.37	1.2927
ig	-1.6075	-98.242	3.3814e-2	76.839	1.4565	1.68	1.4447
ii	2.6218	3.8691e-3	4.3824e-2	93.837	2.1899	2.26	2.1855
ik	-6.5963e-1	-195.90	4.5227e-2	98.627	1.8770	1.65	1.8984
il	-1.0167	-5.7296e-10	4.2627e-2	91.269	2.3385	2.26	2.3462
in	-2.7844	-1.0196	4.0656e-2	94.183	1.6730	1.49	1.7025
ip	-1.8159	-17.771	4.2985e-2	94.162	2.1548	2.4	2.1398
iq	-1.0367	-139.13	4.4404e-2	101.38	1.5374	1.49	1.5412
is	-1.0995	-3.6234e-6	3.7369e-2	86.942	1.4932	1.49	1.4939
it	3.7598	-3.7961	4.1127e-2	93.486	1.5078	1.49	1.5096
iv	3.4996e-1	-23.047	4.1331e-2	89.513	2.0820	2.05	2.0838
iw	-2.1529	-286.15	6.7112e-2	140.95	2.9815	3.05	2.9605
la	-2.0191	-115.11	3.6333e-2	80.904	1.6620	1.72	1.6587
lf	-2.4719e-2	1.7126e-2	5.3595e-2	113.46	2.7901	2.75	2.7939
lg	-1.7217	-98.393	3.3773e-2	76.937	1.4429	1.72	1.4283
ll	-8.3489e-1	-3.1146	4.3037e-2	91.962	2.3553	2.35	2.3560
lw	-1.8671	-121.52	6.7639e-2	142.26	3.2698	3.4	3.2477
ly	7.6486	-4.0492e-3	5.4647e-2	118.35	2.1727	2.46	2.1209

pa	3.2160	-9.2059e-8	3.7045e-2	85.185	1.3599	1.32	1.3635
pf	4.9661	34.671	5.7527e-2	122.22	2.6968	2.8	2.6870
pi	7.7012e-1	-9.1557e-1	4.507e-2	97.652	2.2170	2.33	2.2126
pl	3.2054	-4.8344e-1	4.465e-2	97.078	2.0452	2.22	2.0372
py	-2.2559	-296.69	5.877e-2	128.90	2.1194	1.8	2.1965
sl	1.2870	-3.0003e-2	3.6832e-2	85.491	1.3794	1.49	1.3713
va	6.5104	2.3881e-1	3.367e-2	76.386	1.2239	1.16	1.2375
vg	-2.7431	-82.813	3.0927e-2	71.461	1.3669	1.19	1.3800
vl	1.4568	1.162e-2	4.0159e-2	86.943	2.0322	2	2.0343
vv	7.2806e-1	-16.073	3.8668e-2	84.714	1.8962	1.71	1.9053
ww	3.9921	-1.9191e-1	8.3146e-2	175.70	3.7002	3.6	3.7448
yl	-1.8885	-297.83	5.7653e-2	124.96	2.2089	2.4	2.1657

r2(Y,Y_EST) = 0.940321832704529
r2(Y,Y_PRED) = 0.9238

---

r2 = 0.9400
id = 155
Indices = 4
Molecules = 47
r2(iAPMLHg) = 0.526626389497201; dx2=6.15920592729742E-05
r2(imDrDQg) = 0.0249804427387664; dx2=0.000512616418713192
r2(IHMdpMg) = 0.739782021807653; dx2=0.000128347121604344
r2(IHMdOMg) = 0.662751528392356; dx2=495.687834122231
y = 2.628600269530879E+000+-26.705*iAPMLHg+-5.8934*imDrDQg+215.60*IHMdpMg+-9.2978e-2*IHMdOMg

Mol	iAPMLHg	imDrDQg	IHMdpMg	IHMdOMg	Y_est	Y	Y_pred
af	3.0543e-2	7.9058e-2	4.7487e-2	103.33	1.9778	1.72	2.0010
al	3.7823e-2	3.8518e-2	3.5221e-2	78.283	1.7065	1.7	1.7069
av	5.6554e-2	2.1113e-2	3.6111e-2	79.717	1.3675	1.16	1.4726
ff	3.3726e-2	4.2605e-2	6.3041e-2	130.03	2.9785	3.1	2.9412
fg	3.3608e-2	4.6781e-2	4.6636e-2	103.97	1.8431	1.77	1.8457
fl	3.1299e-2	4.0745e-2	5.2769e-2	110.11	2.6918	2.87	2.6723
fp	3.0186e-2	2.7926e-2	5.6856e-2	121.93	2.5792	2.7	2.5717
fy	2.4496e-2	1.7546e-2	6.6563e-2	142.77	2.9475	3.13	2.9163
gf	2.8565e-2	5.1031e-2	4.5237e-2	100.37	1.9859	1.8	1.9937
gi	4.0207e-2	4.6058e-2	3.3567e-2	75.788	1.4738	1.7	1.4618
gl	3.7393e-2	4.5865e-2	3.3676e-2	76.133	1.5415	1.68	1.5342
gp	3.5995e-2	1.7116e-2	3.2627e-2	77.719	1.3747	1.35	1.3792
gv	4.5748e-2	4.4838e-2	3.0748e-2	70.898	1.1799	1.13	1.1846
gw	2.5339e-2	5.9861e-2	5.321e-2	117.27	2.1676	1.89	2.1849
gy	2.7973e-2	7.5014e-2	4.6355e-2	105.61	1.6142	1.77	1.5959
ia	4.0379e-2	5.2966e-2	3.6478e-2	81.374	1.5367	1.68	1.5282
i_d	3.1746e-2	4.5266e-2	4.1731e-2	98.727	1.3318	1.37	1.3242
ie	3.4041e-2	3.4705e-2	4.381e-2	102.41	1.4385	1.37	1.4511
ig	3.5318e-2	4.987e-2	3.3814e-2	76.839	1.5374	1.68	1.5288
ii	3.2985e-2	3.135e-2	4.3824e-2	93.837	2.2866	2.26	2.2883
ik	2.824e-2	8.5913e-2	4.5227e-2	98.627	1.9489	1.65	1.9857
il	2.5761e-2	4.7571e-2	4.2627e-2	91.269	2.3646	2.26	2.3810
in	3.8606e-2	7.7294e-3	4.0656e-2	94.183	1.5605	1.49	1.5728

ip	2.5316e-2	5.3701e-2	4.2985e-2	94.162	2.1486	2.4	2.1213
iq	3.3071e-2	8.4079e-2	4.4404e-2	101.38	1.3973	1.49	1.3841
is	4.0311e-2	4.4672e-2	3.7369e-2	86.942	1.2618	1.49	1.2447
it	3.6925e-2	5.3811e-2	4.1127e-2	93.486	1.5002	1.49	1.5007
iv	3.7322e-2	7.1096e-3	4.1331e-2	89.513	2.1782	2.05	2.1909
iw	1.5091e-2	8.7957e-2	6.7112e-2	140.95	3.0713	3.05	3.0764
la	2.9182e-2	5.4295e-2	3.6333e-2	80.904	1.8404	1.72	1.8551
lf	2.9507e-2	2.4795e-2	5.3595e-2	113.46	2.7002	2.75	2.6967
lg	3.0574e-2	4.8331e-2	3.3773e-2	76.937	1.6552	1.72	1.6473
ll	3.1483e-2	4.2438e-2	4.3037e-2	91.962	2.2660	2.35	2.2595
lw	1.8255e-2	2.3004e-2	6.7639e-2	142.26	3.3614	3.4	3.3544
ly	2.9039e-2	3.0925e-3	5.4647e-2	118.35	2.6128	2.46	2.6334
pa	4.216e-2	5.9482e-2	3.7045e-2	85.185	1.2187	1.32	1.2091
pf	2.7226e-2	2.4947e-2	5.7527e-2	122.22	2.7935	2.8	2.7929
pi	3.3992e-2	7.3251e-3	4.507e-2	97.652	2.3152	2.33	2.3139
pl	3.3154e-2	1.6111e-2	4.465e-2	97.078	2.2486	2.22	2.2505
py	2.6142e-2	8.917e-2	5.877e-2	128.90	2.0909	1.8	2.1474
sl	4.3465e-2	5.1367e-2	3.6832e-2	85.491	1.1573	1.49	1.1214
va	4.7199e-2	2.9251e-3	3.367e-2	76.386	1.5079	1.16	1.5599
vg	3.81e-2	6.0256e-2	3.0927e-2	71.461	1.2795	1.19	1.2866
vl	3.8638e-2	3.0905e-2	4.0159e-2	86.943	1.9891	2	1.9884
vv	4.1177e-2	5.2786e-2	3.8668e-2	84.714	1.6781	1.71	1.6754
ww	1.6157e-2	3.9109e-2	8.3146e-2	175.70	3.5566	3.6	3.5390
yl	2.3441e-2	7.795e-2	5.7653e-2	124.96	2.3546	2.4	2.3499

r2(Y,Y_EST) = 0.939986608451982
r2(Y,Y_PRED) = 0.9231

---

r2 = 0.9374
id = 154
Indices = 4
Molecules = 47
r2(iAPMLHg) = 0.526626389497201; dx2=6.15920592729742E-05
r2(aAPrwQt) = 0.0450719773665295; dx2=624920.577456302
r2(IHMdoMg) = 0.725860453578758; dx2=0.0199607572105026
r2(IHMdOCg) = 0.697527518376361; dx2=3.12540374377546
y = 2.021528030038608E+000+-21.196*iAPMLHg+-1.683e-4*aAPrwQt+45.875*IHMdoMg+-3.4049*IHMdOCg

Mol	iAPMLHg	aAPrwQt	IHMdoMg	IHMdOCg	Y_est	Y	Y_pred
af	3.0543e-2	185.19	6.045e-1	7.8854	2.2254	1.72	2.2407
al	3.7823e-2	3.0945	4.495e-1	5.9194	1.6851	1.7	1.6845
av	5.6554e-2	2.7795	4.6046e-1	6.0479	1.3534	1.16	1.4551
ff	3.3726e-2	17.553	7.8347e-1	10.068	2.9648	3.1	2.9368
fg	3.3608e-2	142.87	5.9261e-1	7.8335	1.7988	1.77	1.8006
fl	3.1299e-2	30.459	6.632e-1	8.5190	2.7709	2.87	2.7602
fp	3.0186e-2	143.77	7.2034e-1	9.3496	2.5686	2.7	2.5599
fy	2.4496e-2	43.332	8.441e-1	10.950	2.9344	3.13	2.8975
gf	2.8565e-2	160.22	5.757e-1	7.5828	1.9806	1.8	1.9908
gi	4.0207e-2	74.202	4.2968e-1	5.6956	1.4754	1.7	1.4641
gl	3.7393e-2	74.477	4.3158e-1	5.7214	1.5343	1.68	1.5268

gp	3.5995e-2	46.744	4.2287e-1	5.7108	1.2051	1.35	1.1921
gv	4.5748e-2	60.411	3.9552e-1	5.2842	1.1939	1.13	1.1998
gw	2.5339e-2	1384.7	6.7659e-1	8.8792	2.0571	1.89	2.0716
gy	2.7973e-2	562.26	5.9292e-1	7.8879	1.6766	1.77	1.6658
ia	4.0379e-2	88.004	4.6722e-1	6.1553	1.6263	1.68	1.6240
i_d	3.1746e-2	20.950	5.4071e-1	7.2872	1.3379	1.37	1.3313
ie	3.4041e-2	23.602	5.6593e-1	7.5907	1.4124	1.37	1.4206
ig	3.5318e-2	74.526	4.3386e-1	5.7645	1.5361	1.68	1.5276
ii	3.2985e-2	14.504	5.5722e-1	7.2193	2.3014	2.26	2.3042
ik	2.824e-2	4018.1	5.7673e-1	7.5237	1.5867	1.65	1.4851
il	2.5761e-2	28.625	5.4029e-1	7.0083	2.3939	2.26	2.4158
in	3.8606e-2	568.65	5.2621e-1	7.0196	1.3463	1.49	1.3347
ip	2.5316e-2	413.54	5.4923e-1	7.1746	2.1824	2.4	2.1584
iq	3.3071e-2	238.26	5.7128e-1	7.5909	1.6416	1.49	1.6530
is	4.0311e-2	21.938	4.8334e-1	6.4667	1.3181	1.49	1.3061
it	3.6925e-2	26.092	5.3133e-1	7.0292	1.6755	1.49	1.6826
iv	3.7322e-2	11.897	5.2658e-1	6.8542	2.0474	2.05	2.0473
iw	1.5091e-2	1025.3	8.4733e-1	10.904	3.2733	3.05	3.3106
la	2.9182e-2	90.105	4.6526e-1	6.1251	1.8762	1.72	1.8958
lf	2.9507e-2	30.037	6.8172e-1	8.7732	2.7930	2.75	2.7969
lg	3.0574e-2	74.225	4.3438e-1	5.7716	1.6364	1.72	1.6264
ll	3.1483e-2	22.456	5.4535e-1	7.0683	2.3015	2.35	2.2977
lw	1.8255e-2	822.11	8.5577e-1	11.010	3.2667	3.4	3.2484
ly	2.9039e-2	20.767	6.9913e-1	9.0738	2.5797	2.46	2.5869
pa	4.216e-2	23.142	4.7723e-1	6.3587	1.3661	1.32	1.3694
pf	2.7226e-2	3.8371	7.294e-1	9.4226	2.8220	2.8	2.8238
pi	3.3992e-2	7.4210	5.7407e-1	7.4711	2.1968	2.33	2.1918
pl	3.3154e-2	868.98	5.6982e-1	7.4207	2.0462	2.22	2.0380
py	2.6142e-2	2002.1	7.4858e-1	9.7957	2.1181	1.8	2.1706
sl	4.3465e-2	21.670	4.7508e-1	6.3554	1.2513	1.49	1.2293
va	4.7199e-2	1.5930	4.3283e-1	5.7395	1.3344	1.16	1.3541
vg	3.81e-2	64.395	3.9771e-1	5.3216	1.3285	1.19	1.3380
vl	3.8638e-2	21.385	5.1049e-1	6.6492	1.9778	2	1.9765
vv	4.1177e-2	8.9782	4.936e-1	6.4556	1.8104	1.71	1.8170
ww	1.6157e-2	2571.6	1.0577	13.593	3.4854	3.6	3.4299
yl	2.3441e-2	2059.7	7.336e-1	9.5501	2.3147	2.4	2.3036

r2(Y,Y_EST) = 0.937326659215774
r2(Y,Y_PRED) = 0.9215

---

r2 = 0.8873
id = 153
Indices = 2
Molecules = 47
r2(IHMdoMg) = 0.725860453578758; dx2=0.0199607572105026
r2(IHMdOCg) = 0.697527518376361; dx2=3.12540374377546
y = 1.03185675398204E+000+48.174*IHMdoMg+-3.5509*IHMdOCg

Mol	IHMdoMg	IHMdOCg	Y_est	Y	Y_pred
af	6.045e-1	7.8854	2.1527	1.72	2.1632

al	4.495e-1	5.9194	1.6668	1.7	1.6652
av	4.6046e-1	6.0479	1.7385	1.16	1.7687
ff	7.8347e-1	10.068	3.0242	3.1	3.0168
fg	5.9261e-1	7.8335	1.7642	1.77	1.7641
fl	6.632e-1	8.5190	2.7307	2.87	2.7171
fp	7.2034e-1	9.3496	2.5340	2.7	2.5267
fy	8.441e-1	10.950	2.8131	3.13	2.7754
gf	5.757e-1	7.5828	1.8398	1.8	1.8411
gi	4.2968e-1	5.6956	1.5067	1.7	1.4972
gl	4.3158e-1	5.7214	1.5066	1.68	1.4983
gp	4.2287e-1	5.7108	1.1247	1.35	1.1072
gv	3.9552e-1	5.2842	1.3219	1.13	1.3338
gw	6.7659e-1	8.8792	2.0967	1.89	2.1087
gy	5.9292e-1	7.8879	1.5860	1.77	1.5677
ia	4.6722e-1	6.1553	1.6828	1.68	1.6830
i_d	5.4071e-1	7.2872	1.2039	1.37	1.1723
ie	5.6593e-1	7.5907	1.3411	1.37	1.3366
ig	4.3386e-1	5.7645	1.4634	1.68	1.4536
ii	5.5722e-1	7.2193	2.2403	2.26	2.2391
ik	5.7673e-1	7.5237	2.0993	1.65	2.1114
il	5.4029e-1	7.0083	2.1740	2.26	2.1685
in	5.2621e-1	7.0196	1.4555	1.49	1.4535
ip	5.4923e-1	7.1746	2.0141	2.4	2.0028
iq	5.7128e-1	7.5909	1.5981	1.49	1.6059
is	4.8334e-1	6.4667	1.3536	1.49	1.3456
it	5.3133e-1	7.0292	1.6681	1.49	1.6738
iv	5.2658e-1	6.8542	2.0607	2.05	2.0613
iw	8.4733e-1	10.904	3.1321	3.05	3.1415
la	4.6526e-1	6.1251	1.6956	1.72	1.6947
lf	6.8172e-1	8.7732	2.7202	2.75	2.7179
lg	4.3438e-1	5.7716	1.4633	1.72	1.4516
ll	5.4535e-1	7.0683	2.2047	2.35	2.1951
lw	8.5577e-1	11.010	3.1623	3.4	3.1335
ly	6.9913e-1	9.0738	2.4915	2.46	2.4928
pa	4.7723e-1	6.3587	1.4428	1.32	1.4482
pf	7.294e-1	9.4226	2.7112	2.8	2.7062
pi	5.7407e-1	7.4711	2.1579	2.33	2.1521
pl	5.6982e-1	7.4207	2.1322	2.22	2.1293
py	7.4858e-1	9.7957	2.3103	1.8	2.3585
sl	4.7508e-1	6.3554	1.3509	1.49	1.3433
va	4.3283e-1	5.7395	1.5026	1.16	1.5191
vg	3.9771e-1	5.3216	1.2946	1.19	1.3010
vl	5.1049e-1	6.6492	2.0135	2	2.0143
vv	4.936e-1	6.4556	1.8873	1.71	1.8958
ww	1.0577	13.593	3.7181	3.6	3.7610

yl	7.336e-1	9.5501	2.4608	2.4	2.4641

r2(Y,Y_EST) = 0.887338630053066
r2(Y,Y_PRED) = 0.8731

---

r2 = 0.8837
id = 152
Indices = 2
Molecules = 47
r2(IHMdpMg) = 0.739782021807653; dx2=0.000128347121604344
r2(IHMdOMg) = 0.662751528392356; dx2=495.687834122231
y = 9.353880712342770E-001+240.54*IHMdpMg+-9.8778e-2*IHMdOMg

Mol	IHMdpMg	IHMdOMg	Y_est	Y	Y_pred
af	4.7487e-2	103.33	2.1511	1.72	2.1615
al	3.5221e-2	78.283	1.6748	1.7	1.6736
av	3.6111e-2	79.717	1.7472	1.16	1.7786
ff	6.3041e-2	130.03	3.2551	3.1	3.2805
fg	4.6636e-2	103.97	1.8832	1.77	1.8870
fl	5.2769e-2	110.11	2.7519	2.87	2.7396
fp	5.6856e-2	121.93	2.5675	2.7	2.5615
fy	6.6563e-2	142.77	2.8439	3.13	2.8106
gf	4.5237e-2	100.37	1.9023	1.8	1.9049
gi	3.3567e-2	75.788	1.5234	1.7	1.5146
gl	3.3676e-2	76.133	1.5155	1.68	1.5076
gp	3.2627e-2	77.719	1.1065	1.35	1.0864
gv	3.0748e-2	70.898	1.3283	1.13	1.3406
gw	5.321e-2	117.27	2.1508	1.89	2.1640
gy	4.6355e-2	105.61	1.6536	1.77	1.6442
ia	3.6478e-2	81.374	1.6718	1.68	1.6716
i_d	4.1731e-2	98.727	1.2213	1.37	1.1939
ie	4.381e-2	102.41	1.3575	1.37	1.3558
ig	3.3814e-2	76.839	1.4790	1.68	1.4698
ii	4.3824e-2	93.837	2.2077	2.26	2.2048
ik	4.5227e-2	98.627	2.0721	1.65	2.0827
il	4.2627e-2	91.269	2.1735	2.26	2.1680
in	4.0656e-2	94.183	1.4115	1.49	1.4056
ip	4.2985e-2	94.162	1.9738	2.4	1.9625
iq	4.4404e-2	101.38	1.6022	1.49	1.6101
is	3.7369e-2	86.942	1.3361	1.49	1.3265
it	4.1127e-2	93.486	1.5937	1.49	1.5980
iv	4.1331e-2	89.513	2.0352	2.05	2.0345
iw	6.7112e-2	140.95	3.1557	3.05	3.1681
la	3.6333e-2	80.904	1.6833	1.72	1.6819
lf	5.3595e-2	113.46	2.6197	2.75	2.6123
lg	3.3773e-2	76.937	1.4594	1.72	1.4476
ll	4.3037e-2	91.962	2.2036	2.35	2.1939
lw	6.7639e-2	142.26	3.1531	3.4	3.1233
ly	5.4647e-2	118.35	2.3898	2.46	2.3872
pa	3.7045e-2	85.185	1.4317	1.32	1.4369

pf	5.7527e-2	122.22	2.7002	2.8	2.6947
pi	4.507e-2	97.652	2.1306	2.33	2.1244
pl	4.465e-2	97.078	2.0863	2.22	2.0825
py	5.877e-2	128.90	2.3394	1.8	2.3874
sl	3.6832e-2	85.491	1.3503	1.49	1.3425
va	3.367e-2	76.386	1.4891	1.16	1.5049
vg	3.0927e-2	71.461	1.3157	1.19	1.3234
vl	4.0159e-2	86.943	2.0071	2	2.0076
vv	3.8668e-2	84.714	1.8687	1.71	1.8760
ww	8.3146e-2	175.70	3.5800	3.6	3.5723
yl	5.7653e-2	124.96	2.4599	2.4	2.4631

r2(Y,Y_EST) = 0.883635905971168
r2(Y,Y_PRED) = 0.8696

var	id	r^2	$r^2_{cv(loo)}$	r^2-$r^2_{cv(loo)}$	CrossCor
4	156	0.940321832704529	0.923756497853518	0.016565334851011	0.27718 0.05871 0.04890 0.01138 0.01666 0.98833
4	155	0.939986608451982	0.923076637900501	0.0169099705514814	0.07846 0.58707 0.58933 0.00231 0.00468 0.98833
4	154	0.937326659215774	0.921541211239033	0.0157854479767405	0.22874 0.59023 0.59101 0.20964 0.21348 0.99839
2	153	0.887338630053066	0.873057533034213	0.0142810970188524	0.99839
2	152	0.883635905971168	0.869612894448512	0.0140230115226566	0.98833
1	0	0.809790498328748	0.792067788159534	0.017722710169214	1

Sursa aplicațiilor de mai sus este redată în continuare. Următoarele aplicații au fost folosite (în această ordine este redat codul sursă):

- k_browse_or_query.php
- h_mdf_browse.php
- i_mdf_query.php

```
<?
$definition_page["auto_index"]=FALSE;
$definition_page["copyright"]=array("February 2007","lori");
include("/usr/home/www/data/definitions.php");
?>
Browse or Query MDF SARs by sets.
<BR>
<?
include("0_mdf_definitions.php");

if(array_key_exists("database",$_GET)){
	if($_GET["database"]==server_db_finaliz){
	}elseif($_GET["database"]==server_db_in_work){
	}else{
		$_GET["database"]=server_db_finaliz;
	}
}else{
	$_GET["database"]=server_db_finaliz;
}
```

```
$q=mysql_query("USE `".$_GET["database"]."`");
if(!$q){
      echo("dB error<br>\r\n");
      echo(mysql_errno($mysql_link)."<br>\r\n");
      echo(mysql_error($mysql_link)."<br>\r\n");
      die();
}
$q=mysql_query("SELECT DISTINCT `name` FROM `".server_table_ress."`");
if(!$q){
      echo("query error<br>\r\n");
      echo(mysql_errno($mysql_link)."<br>\r\n");
      echo(mysql_error($mysql_link)."<br>\r\n");
      die();
}
for(;$r=mysql_fetch_row($q);){
      $set[]=$r[0];
}
?>
<table border='1' width='90%'>
<tr>
<td align='center' valign='top'>
<form method='get' action='h_mdf_browse.php'>
<table width='90%'>
<tr align='center'>
<td>
Browse
<td>
<input type='text' readonly='true' name='database' value='<?
echo($_GET['database']); ?>'>
<tr align='center'>
<td>
<select name='set'>
<?
 for($i=0;$i<count($set);$i++)
  echo("<option value='".$set[$i]."'>".$set[$i]);
?>
</select>
<td>
<input type='submit'>
</table>
</form>
<td align='center' valign='top'>
<form method='get' action='i_mdf_query.php'>
<table width='90%'>
<tr align='center'>
<td>
Query
<td>
<input type='text' readonly='true' name='database' value='<?
echo($_GET['database']); ?>'>
<tr align='center'>
<td>
<select name='set'>
<?
 for($i=0;$i<count($set);$i++)
  echo("<option value='".$set[$i]."'>".$set[$i]);
echo("</select><td><input type='submit'></table></form></table>");
?>
<?
include("0_mdf_definitions.php");
if(array_key_exists("database",$_GET)){
      if($_GET["database"]==server_db_finaliz){
      }elseif($_GET["database"]==server_db_in_work){
```

```
      }else{
            $_GET["database"]=server_db_finaliz;
      }
}else{
      $_GET["database"]=server_db_finaliz;
}
$q=mysql_query("USE `".$_GET["database"]."`");
if(!$q){
      echo("dB error<br>\r\n");
      echo(mysql_errno($mysql_link)."<br>\r\n");
      echo(mysql_error($mysql_link)."<br>\r\n");
      die();
}
 echo($_GET['set']);
 if(!$_GET['set']) echo('No data set=.');
 $q=mysql_query("SELECT `n`,`r` FROM `".$_GET['set'].server_table_yval."` ORDER
BY `r` DESC LIMIT 1");
 $r=mysql_fetch_row($q);
 mysql_free_result($q);
 echo("<table border=1>");
 echo("<tr>");
 echo("<td>0");
 echo("<td>y=a+".$r[0]."*b");
 echo("<td>".$r[1]);
 $q=mysql_query("SELECT * FROM `".server_table_ress."` WHERE `name` LIKE
'".$_GET['set']."' ORDER BY `id` ASC");
 $l_t[0]=$r[0];
 while($r=mysql_fetch_row($q)){
  echo("<tr>");
  echo("<td>".$r[0]);
  for($i=2;$i<count($r);$i++)
   echo("<td>".str_replace("\r\n","",$r[$i]));
  list($z,$tmp)=explode("=",$r[2],2);
  $tmp=explode("*",$tmp);
  $j=strrpos($tmp[0],"+");
  $l_t[]=substr($tmp[0],$j+1);
  for($j=1;$j<count($tmp);$j++){
   $j=strrpos($tmp[0],"+");
   $l_t[]=substr($tmp[0],$j+1);
  }
 }
 echo("</table>\r\n");
 mysql_free_result($q);
 exit();
 $tmp=file_get_contents("rez.txt");
 $z=explode("\r\n",$tmp);
 unset($tmp);
 for($i=0;$i<count($z)-1;$i++){
  $tmp=explode(" ",$z[$i]);
  for($j=0;$j<3;$j++){
   if (nu_este($l_t,$tmp[$j])){
    $l_t[]=$tmp[$j];
   }
  }
  unset($tmp);
 }
 for($i=0;$i<count($l_t);$i++)
  echo($l_t[$i]."<BR>\r\n");
function nu_este(&$lista,&$element){
 $este=1;
 for($i=0;$i<count($lista);$i++)
  if ($lista[$i]==$element) $este=0;
 return $este;
}
```

```
?>
<?
 $file=file_get_contents("y_names.txt");
 $n1=explode("\r\n",$file);
 unset($file);
 if(!$n1[count($n1)-1]) unset($n1[count($n1)-1]);
include("0_mdf_definitions.php");
if(array_key_exists("database",$_GET)){
      if($_GET["database"]==server_db_finaliz){
      }elseif($_GET["database"]==server_db_in_work){
      }else{
            $_GET["database"]=server_db_finaliz;
      }
}else{
      $_GET["database"]=server_db_finaliz;
}
$q=mysql_query("USE `".$_GET["database"]."`");
if(!$q){
      echo("dB error<br>\r\n");
      echo(mysql_errno($mysql_link)."<br>\r\n");
      echo(mysql_error($mysql_link)."<br>\r\n");
      die();
}
 $sett = "";
 if (array_key_exists("set",$_GET)) $sett="WHERE `name` LIKE '".$_GET['set']."'
";
 $q=mysql_query("SELECT DISTINCT `name` FROM `".server_table_ress."`
".$sett."ORDER BY `name`");
 $i=0;
 while($row=mysql_fetch_row($q)) $name_set[$i++]=$row[0];
 if(!isset($name_set)) exit("no data\r\n");
 mysql_free_result($q);
 for($i=0;$i<count($name_set);$i++){
  $q=mysql_query("SELECT * FROM `".$name_set[$i].server_table_data."`");
  $j=0;
  if($q){
   while($r=mysql_fetch_array($q)) $y[$j++]=$r[0];
   mysql_free_result($q);
  }
  $q=mysql_query("SHOW COLUMNS FROM `".$name_set[$i].server_table_tmpx."`");
  $j=0;
  if($q){
   while($r=mysql_fetch_array($q)) $m[$j++]=$r[0];
   mysql_free_result($q);
  }
  unset($m[count($m)-1]);
  echo("<HR size='8' color='black' bgcolor='black'>\r\n");
  echo("<B>".$name_set[$i]."<BR>\r\nMDF Size: ");
  echo(count($n1));
  $q=mysql_query("SELECT `n`,`r` FROM `".$name_set[$i].server_table_yval."`
ORDER BY `r` DESC LIMIT 1");
  $row=mysql_fetch_row($q);
  echo("<BR>\r\ndy2=".(m2($y,$y)-m1($y)*m1($y)));
  mysql_free_result($q);
  echo("<BR>\r\nr2(".$row[0].") =
".sprintf("%.4f",spalare($row[1]))."</B><BR>\r\n");
  $este=$row[0][0];
  $d[1]=substr($row[0],1);
  if($este=="I") $t[1]=0;
  if($este=="i") $t[1]=1;
  if($este=="A") $t[1]=2;
  if($este=="a") $t[1]=3;
  if($este=="L") $t[1]=4;
  if($este=="l") $t[1]=5;
```

```
  $row[2]=array_search($d[1],$n1)+1;
  $q=mysql_query("SELECT * FROM `".$name_set[$i].server_table_tmpx."` WHERE
`id`='".$row[2]."' LIMIT 1");
  if($q){
   $x[0]=mysql_fetch_row($q);
   unset($x[0][count($x[0])-1]);
   mysql_free_result($q);
   fn_x($x[0],$t[1]);
   for($l=0;$l<count($x[0]);$l++)
    $x[0][$l]=spalare($x[0][$l]);
  }
  unset($c);
  echo("dx2=".(m2($x[0],$x[0])-m1($x[0])*m1($x[0]))."<BR>\r\n");
  get_coefs($y,$x,$c);
  $c[0]=spalare($c[0]);
  $c[1]=spalare($c[1]);
  $x[1]=$x[0];
  unset($x[0]);
  calc_eq($c,$x,$y_pred);
  new_y($y,$x,$y_est);
  echo(put_coef_ind($c,$d,$t)."<BR>");
  echo("<table border=1>");
  echo("<tr>");
  echo("<td>Mol");
  echo("<td>".name_ind($d[1],$t[1]));
  echo("<td>Y_est");
  echo("<td>Y");
  echo("<td>Y_pred");
  for($l=0;$l<count($x[1]);$l++){
   echo("<tr>");
   echo("<td>".$m[$l]);
   echo("<td>".$x[1][$l]);
   echo("<td>".$y_pred[$l]);
   echo("<td>".$y[$l]);
   echo("<td>".$y_est[$l]);
  }
  echo("</table>");
  $r0=coef_r($y,$y_pred);
  $r_est0=coef_r($y,$y_est);
  echo("r2(Y,Y_EST) = ".sprintf("%.4f",$r0)."<BR>\r\n");
  echo("r2(Y,Y_PRED) = ".sprintf("%.4f",$r_est0)."<BR>\r\n");
  unset($y_new);
  echo("<HR size='8' color='black' bgcolor='black'>\r\n");
  unset($d);
  unset($t);
  unset($x);
  unset($y_est);
  unset($y_pred);
  unset($c);
  $q=mysql_query("SELECT * FROM `".server_table_ress."` WHERE `name` LIKE
'".$name_set[$i]."' ORDER BY `r` DESC LIMIT 30");
  $j=0;
  while($row=mysql_fetch_row($q)){
   $id[$j]=$row[0];
   if(!(strpos($row[2],"=")))die();
   list($row[0],$eq)=explode("=",$row[2],2);
   get_coef_ind($eq,$c[$j],$d[$j]);
   $r[$j]=$row[3];
   $j++;
  }
  mysql_free_result($q);
  for($j=0;$j<count($d);$j++)
   for($k=1;$k<count($d[$j]);$k++){
    $este=$d[$j][$k][0];
```

```
    $d[$j][$k]=substr($d[$j][$k],1);
    if($este=="I") $t[$j][$k]=0;
    if($este=="i") $t[$j][$k]=1;
    if($este=="A") $t[$j][$k]=2;
    if($este=="a") $t[$j][$k]=3;
    if($este=="L") $t[$j][$k]=4;
    if($este=="l") $t[$j][$k]=5;
    $p[$j][$k]=array_search($d[$j][$k],$n1)+1;
  }
  for($j=0;$j<count($d);$j++){//a cata inregistrare din mol_top
   echo("r2 = ".sprintf("%.4f",$r[$j])."<BR>");
   echo("id = ".$id[$j]."<BR>");
   echo("Indices = ".(count($d[$j])-1)."<BR>");
   echo("Molecules = ".count($y)."<BR>");
   for($k=1;$k<count($d[$j]);$k++){//al catelea indice din lista
    if($t[$j][$k]==0) $xx_n="I";
    if($t[$j][$k]==1) $xx_n="i";
    if($t[$j][$k]==2) $xx_n="A";
    if($t[$j][$k]==3) $xx_n="a";
    if($t[$j][$k]==4) $xx_n="L";
    if($t[$j][$k]==5) $xx_n="l";
    $xx_n.=$d[$j][$k];
    $q=mysql_query("SELECT `r` FROM `".$name_set[$i].server_table_yval."` WHERE
`n` REGEXP BINARY '".$xx_n."' LIMIT 1");
    $af=mysql_fetch_row($q);
    mysql_free_result($q);
    echo("r2(".$xx_n.") = ".$af[0]."; ");
    unset($af);
    unset($xx_n);
    $q=mysql_query("SELECT * FROM `".$name_set[$i].server_table_tmpx."` WHERE
`id`='".$p[$j][$k]."' LIMIT 1");
    if($q){
     $x[$j][$k]=mysql_fetch_row($q);
     unset($x[$j][$k][count($x[$j][$k])-1]);
     mysql_free_result($q);
     fn_x($x[$j][$k],$t[$j][$k]);
     for($l=0;$l<count($x[$j][$k]);$l++)
      $x[$j][$k][$l]=spalare($x[$j][$k][$l]);
    }
    echo("dx2=".(m2($x[$j][$k],$x[$j][$k])-
m1($x[$j][$k])*m1($x[$j][$k]))."<BR>\r\n");
   }
   $cr[$j]="";
   for($k=1;$k<count($d[$j]);$k++)
    for($l=$k+1;$l<count($d[$j]);$l++)
     $cr[$j].=sprintf("%.5f",coef_r($x[$j][$k],$x[$j][$l]))." ";
   calc_eq($c[$j],$x[$j],$y_pred);
   new_y($y,$x[$j],$y_est);
   echo(put_coef_ind($c[$j],$d[$j],$t[$j])."<BR>");
   echo("<table border=1>");
   echo("<tr>");
   echo("<td>Mol");
   for($k=1;$k<count($d[$j]);$k++)
    echo("<td>".name_ind($d[$j][$k],$t[$j][$k]));
   echo("<td>Y_est");
   echo("<td>Y");
   echo("<td>Y_pred");
   for($l=0;$l<count($x[$j][1]);$l++){
    echo("<tr>");
    echo("<td>".$m[$l]);
    for($k=1;$k<count($d[$j]);$k++)
     echo("<td>".$x[$j][$k][$l]);
    echo("<td>".$y_pred[$l]);
    echo("<td>".$y[$l]);
```

```
    echo("<td>".$y_est[$l]);
   }
   echo("</table>");
   $r[$j]=coef_r($y,$y_pred);
   $r_est[$j]=coef_r($y,$y_est);
   echo("r2(Y,Y_EST) =
".$r[$j]."<BR>\r\n");//sprintf("%.4f",$r[$j])."<BR>\r\n");
   echo("r2(Y,Y_PRED) = ".sprintf("%.4f",$r_est[$j])."<BR>\r\n");
   unset($y_new);
   echo("<HR size='3' color='black' bgcolor='black'>");
  }
  echo("<HR size='30' color='black' bgcolor='black'>");
  echo("<table border=1>");
echo("<tr><td>var<td>id<td>r<sup>2</sup><td>r<sup>2</sup><sub>cv(loo)</sub><td>r
<sup>2</sup>-r<sup>2</sup><sub>cv(loo)</sub>");
  echo("<td>CrossCor");
  for ($j=0;$j<count($id);$j++){
      echo("<tr>");
      echo("<td>".(count($d[$j])-
1)."<td>".$id[$j]."<td>".$r[$j]."<td>".$r_est[$j]."<td>".($r[$j]-$r_est[$j]));
      echo("<td>".$cr[$j]);
  }
  echo("<tr>");
  echo("<td>1<td>0");
  echo("<td>".$r0."<td>".$r_est0."<td>".($r0-$r_est0)."<td>1");
  echo("</table>");
  unset($cr);
  unset($c);
  unset($d);
  unset($p);
  unset($t);
  unset($x);
  unset($y);
  unset($y_est);
  unset($y_pred);
  unset($m);
  unset($r);
  unset($r_est);
  unset($id);
}
function exclude_i($i,&$data,&$new_data){
  for($j=0;$j<$i;$j++)
   $new_data[$j]=$data[$j];
  for($j=$i+1;$j<count($data);$j++)
   $new_data[$j-1]=$data[$j];
  return $data[$i];
}
function calc_eq(&$coef,&$val,&$rez){//predictie pentru fiecare molecula
 for($i=0;$i<count($val[1]);$i++){//a cata molecula
  $rez[$i]=$coef[0];
  for($j=1;$j<count($coef);$j++){//al catelea indice si coeficient
   $rez[$i]+=$coef[$j]*$val[$j][$i];
  }
  $rez[$i]=spalare($rez[$i]);
 }
}
function calc_eq0(&$coef,&$val,&$rez){//predictie pentru fiecare molecula
 for($i=0;$i<count($val[0]);$i++){//a cata molecula
  $rez[$i]=$coef[0];
  for($j=1;$j<count($coef);$j++){//al catelea indice si coeficient
   $rez[$i]+=$coef[$j]*$val[$j][$i];
  }
  $rez[$i]=spalare($rez[$i]);
 }
```

```
}
function calc_1v(&$coef,&$val,&$rez){//predictie pentru o molecula
 $rez=$coef[0];
 for($j=1;$j<count($coef);$j++){//al catelea indice si coeficient
  $rez+=$coef[$j]*$val[$j-1];
 }
 $rez=spalare($rez);
}
function new_y(&$y_old,&$x_old,&$y_new){//x[ind][mol]
  $cate=count($x_old[1]);
  $cati=count($x_old);
  for($i=0;$i<$cate;$i++){
   $rest[0]=exclude_i($i,$y_old,$y_new_i);
   for($j=1;$j<=$cati;$j++)
    $rest[$j-1]=exclude_i($i,$x_old[$j],$x_new_i[$j-1]);
   get_coefs($y_new_i,$x_new_i,$c_new);
   calc_1v($c_new,$rest,$y_new[$i]);//predictie pentru o molecula
  }
}
function af(&$y,&$x){
 for($i=0;$i<count($y);$i++){
  echo($y[$i]." = ");
  for($j=0;$j<count($x);$j++)
   echo($x[$j][$i].",");
  echo("<BR>");
 }
 echo("<HR>");
}
function get_coefs(&$y,&$x,&$b){
 $b[0] = m1($y);//x[ind][mol]
 $a[0][0] = 1.0;
 for($k=1;$k<=count($x);$k++){
  $a[0][$k]  = m1($x[$k-1]);
  $a[$k][0]  = $a[0][$k];
  $a[$k][$k] = m2($x[$k-1],$x[$k-1]);
  $b[$k] = m2($x[$k-1],$y);
  for($l=$k+1;$l<=count($x);$l++){
   $a[$k][$l] = m2($x[$k-1],$x[$l-1]);
   $a[$l][$k] = $a[$k][$l];
  }
 }
 $a[0][0] = gauss($b,$a);
}
function coef_r(&$y1,&$y2){
  $my1=m1($y1);
  $dy2=m2($y1,$y1)-$my1*$my1;
  $mx1=m1($y2);
  $mxy=m2($y2,$y1);
  $m2x=$mx1*$mx1;
  $mx2=m2($y2,$y2);
  $dx2=$mx2-$m2x;
  return pow($mxy-$mx1*$my1,2)/($dx2*$dy2);
}
function get_coef_ind(&$eq,&$coef,&$indx){
 $indx[0]="1";
 $tmp=explode("*",$eq);
 $n=count($tmp);
 for($i=0;$i<count($tmp)-1;$i++){
  $j=strrpos($tmp[$i],"+");
  $coef[$i]=substr($tmp[$i],0,$j);
  $indx[$i+1]=substr($tmp[$i],$j+1);
 }
 $coef[$n-1]=$tmp[$n-1];
 for($i=1;$i<count($coef);$i++){
```

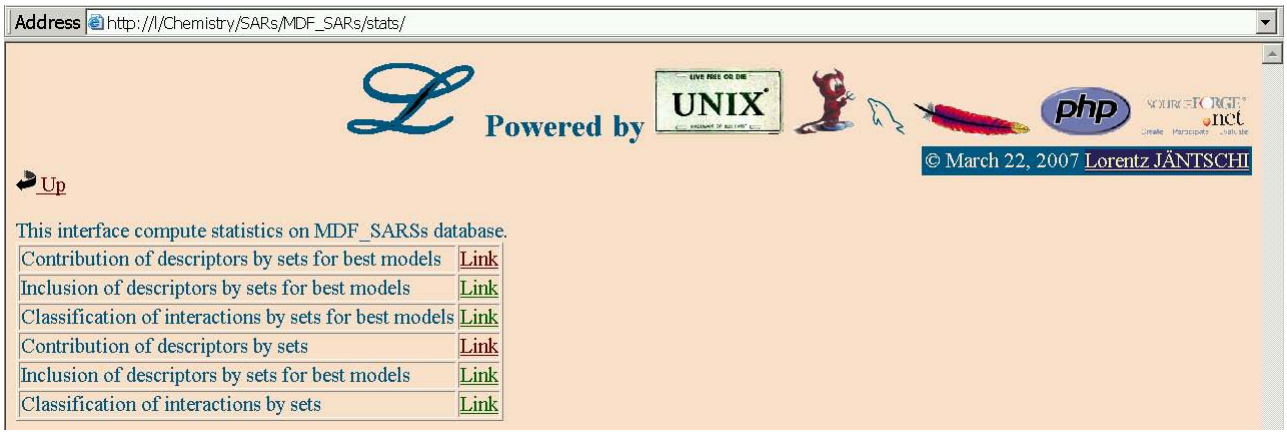
```
  $coef[$i]=spalare($coef[$i]);
 }
}
function put_coef_ind(&$coef,&$indx,$tip){
 $eq="y = ".$coef[0];
 for($i=1;$i<count($coef);$i++){
  $eq.="+".$coef[$i]."*".name_ind($indx[$i],$tip[$i]);
 }
 return $eq;
}
function name_ind(&$ind,&$tip){
  if($tip==0) return "I".$ind;
  if($tip==1) return "i".$ind;
  if($tip==2) return "A".$ind;
  if($tip==3) return "a".$ind;
  if($tip==4) return "L".$ind;
  if($tip==5) return "l".$ind;
  return $ind;
}
function fn_x(&$x,$n){
 for($l=0;$l<count($x);$l++)
  if (!is_numeric($x[$l])) $x[$l] = (float)"INF";
 if($n==1) for($l=0;$l<count($x);$l++) $x[$l]=1.0/$x[$l];
 if($n==2) for($l=0;$l<count($x);$l++) $x[$l]=abs($x[$l]);
 if($n==3) for($l=0;$l<count($x);$l++) $x[$l]=1.0/abs($x[$l]);
 if($n==4) for($l=0;$l<count($x);$l++) $x[$l]=log(abs($x[$l]));
 if($n==5) for($l=0;$l<count($x);$l++) $x[$l]=log($x[$l]);
}
function m1(&$v){
 $rez=0;
 $n=count($v);
 for($i=0;$i<$n;$i++)
  $rez+=$v[$i];
 return $rez/$n;
}
function m2(&$v,&$u){
 $rez=0;
 $n=count($v);
 for($i=0;$i<$n;$i++)
  $rez+=$v[$i]*$u[$i];
 return $rez/$n;
}
function gauss(&$b,&$a){
 for($i=0;$i<count($a);$i++){
  $m=$i;//caut max in coloana
  for($j=$i+1;$j<count($a);$j++)
    if(abs($a[$i][$j])>abs($a[$i][$m]))$m=$j;
  if($a[$i][$m]==0)return -1;
  if($m != $i){//inlocuiesc linii
    $tmp=$b[$m];$b[$m]=$b[$i];$b[$i]=$tmp;
    for($j=0;$j<count($a);$j++){
      $tmp=$a[$m][$j];$a[$m][$j]=$a[$i][$j];$a[$i][$j]=$tmp;
    }
  }
  if(!abs($a[$i][$i])) return -1;
  for($j=$i+1;$j<count($a);$j++)//impart linie
    $a[$i][$j]/=$a[$i][$i];
  $b[$i]/=$a[$i][$i];
  $a[$i][$i]=1;
  for($j=$i+1;$j<count($a);$j++){//fac 0 sub diagonala
    $b[$j]-=$a[$j][$i]*$b[$i];
    for($k=count($a)-1;$k>$i;$k--)
      $a[$j][$k]-=$a[$j][$i]*$a[$i][$k];
  }
```

```
 }
 for($i=count($a)-1;$i>0;$i--)//fac 0 peste diagonala
  for($j=$i-1;$j>=0;$j--)
    $b[$j]-=$a[$j][$i]*$b[$i];
 return 1;
}
function spalare($value){
 if(!is_finite($value))
  return (float)"INF";
 $log_cifre=log(abs($value),10);
 if($log_cifre<0) $cifre=(int)$log_cifre;
 if($log_cifre>0) $cifre=(int)($log_cifre+1);
 $new2value=((int)($value*pow(10,5-$cifre)))*pow(10,-4);
 $new3value=$new2value."e".($cifre-1);
 if(($cifre>0)&&($cifre<6)) $new3value=sprintf("%.".(5-$cifre)."f",$new3value);
 return $new3value;
}
function y_x_coefs(&$xy,$ind,$ll){
 for($i=1;$i<count($xy);$i++){
  for($j=1;$j<count($xy[0])-1;$j++)
   $x_x[$j-1][$i-1]=$xy[$i][$j];
  $x_y[$i-1]=$xy[$i][count($xy[0])-1];
 }
 get_coefs($x_y,$x_x,$x_coef);
echo("y=".sprintf("%1.15f",$x_coef[0]));
 for($i=1;$i<count($x_coef);$i++)
  echo("+".name_ind($ind[$i],$ll[$i])."*".sprintf("%1.15f",$x_coef[$i]));
 echo("<BR>");
}
?>
```

MDF SAR Statistics

Această aplicație este menită să integreze informațiile complexe provenite din investigarea structură-activitate pe toate seturile studiate, şi anume este capabilă să producă statistici pentru contribuția fiecărui descriptor, ratei de apariție a fiecărui parametru în modelel valide, legăturii între activitate şi apariția unui anumit parametru structural al modelului, ş.a.m.d.

Captura de ecran a aplicației este redată mai jos:



Statistica publicațiilor rezultate din derularea proiectului

Statistica publicațiilor rezultate din derularea proiectului este redată mai jos:

- lucrări susținute la conferințe internaționale (20):
  - sub formă de prezentări orale (13):
    - în plen (4): [43], [47], [48], [49]
    - în secțiuni (7): [42], [53], [57], [58], [59], [60], [61]
    - scurte prezentări în plen (2): [44], [45]
  - sub formă de postere (7): [46], [50], [51], [52], [54], [55], [56]
- lucrări publicate în reviste de specialitate (13):
  - reviste indexate Thompson ISI (3): [65], [66], [67]
  - reviste indexate de alte sisteme de indexare (7): [63], [64], [69], [70], [73], [74], [75]
  - reviste clasificate CNCSIS (3):
    - tip B/B+ (2): [68], [71]
    - tip C (1): [72]
- lucrări acceptate spre publicare în reviste de specialitate (3):
  - reviste indexate Thompson ISI (2): [81], [86]
  - reviste indexate de alte sisteme de indexare (1): [76]
- lucrări trimise spre publicare în reviste de specialitate (9):
  - reviste indexate Thompson ISI (8): [77], [78], [79], [80], [82], [83], [84], [85]
  - reviste indexate de alte sisteme de indexare (1): [87]
- cărți publicate (4):
  - indexate internațional (4): [88], [89], [90], [91]
  - recunoscute CNCSIS (2): [89], [90]
- cărți propuse și acceptate spre publicare în sisteme indexate internațional (1): [92]

Cumulând totalul materialelor publicate pe baza rezultatelor obținute prin derularea proiectului rezultă:

- Total publicații apărute: 37 (din care în sisteme indexate internațional: 14);
- Total publicații în curs de apariție: 4 (din care în sisteme indexate internațional: 4);
- Total publicații trimise spre publicare: 9 (din care în sisteme indexate internațional: 9);

Bibliografie (cu includerea publicațiilor rezultate din derularea proiectului)

[1] Computational approaches to structural and functional analysis of plastocyanin and other blue copper proteins, proteins, F. De Rienzo, R. R. Gabdoulline, R. C. Wade, M. Sola, and M. C. Menziani, Cellular and Molecular Life Sciences, Springer-Verlag, Issue: Volume 61, Number 10, Date: May 2004, Pages: 1123 – 1142.

[2] Atomistic Topological Indices Applied to Benzodiazepines using Various Regression Methods, Wikler D. A., Burden F. R., Watkins A. J. R, Quantitative Structure Activity Relationships, Elsevier, 1998, 17(1), p. 14-19.

[3] Computation of the influence of chemical substitution on the pKa of pyridine using semiempirical and ab initio methods, I-Jen Chen and Alexander D. MacKerell Jr, Theoretical Chemistry Accounts: Theory, Computation, and Modeling (Theoretica Chimica Acta), Springer-Verlag, Volume 103, Number 6, Date: April 2000, Pages: 483 – 494.

[4] Development of an Expert System for the Remediation of Petroleum-Contaminated Sites, Z. Chen, G.H. Huang, C.W. Chan, L.Q. Geng and J. Xia, Environmental Modeling and Assessment, Springer Science+Business Media B.V., Volume 8, Number 4, Date: December 2003, Pages: 323 – 334.

[5] CoMFA, HQSAR and molecular docking studies of butitaxel analogues with b-tubulin, Suzanne L. Cunningham, Albert R. Cunningham and Billy W. Day, Journal of Molecular Modeling, Volume 11, Number 1, Date: February 2005, Pages: 48 – 54.

[6] Unity Program for SIMCA (Soft Independent Modeling Class Analogy); Tripos Associates, St. Louis, MO.

[7] Alfred Merz, Didier Rognan, Gerd Folkers, 3D QSAR Study of N2-phenylguanines as Inhibitors of Herpes Simplex Virus Thymide Kinase, Antiviral and Antitumor Research, http://www.pharma.ethz.ch/text/research/tk/qsar.html.

[8] Gurba P. E., Parham M. E., Voltano J. R., Comparison of QSAR Models Developed for Acute Oral Toxicity (LD50) by Regression and Neural Network Techniques, Conference on Computational Methods in Toxicology – April, 1998, Holiday Inn/I-675, Dayton, Ohio, USA, abstract available at http://www.ccl.net/ccl/toxicology/abstracts/abs9.html.

[9] HyperChem, Molecular Modelling System; Hypercube Inc., http://hyper.com/products/Professional/

[10] Molconn-Z, http://www.eslc.vabiotech.com/molconn.

[11] Waller C. L., Wyrick S. D., Park H. M., Kemp W. E., Smith F. T., Conformational Analysis, Molecular Modeling, and Quantitative Structure-Activity Relationship Studies of Agents for the Inhibition of Astrocytic Chloride Transport, Pharm. Res., 1994, 11(1), p. 47-53.

[12] Horwitz J. P., Massova I., Wiese T., Wozniak J., Corbett T. H., Sebolt-Leopold J. S., Capps D. B., Leopold W. R., Comparative Molecular Field Analysis of in Vitro Growth Inhibition of L1210 and HCT-8 Cells by Some Pyrazoloacridines, J. Med. Chem., 1993, 36(23), p. 3511-3516.

[13] McGaughey G. B., MewShaw R. E., Molecular Modeling and the Design of Dopamine D2 Partial Agonists, (presented at the Charleston Conference; march; 1998), submitted in may 1998, Network Science, http://www.netsci.org/Science/Compchem/feature20.html.

[14] Chuman H., Karasawa M., Fujita T., A Novel Three-Dimensional QSAR Procedure: Voronoi Field Analysis, QSAR, 1998, 17(4), p. 313-326.

[15] Walter C. L., Kellogg G. E., Adding Chemical Information of CoMFA Models with Alternative 3D QSAR Fields.

[16] Merz A., Rognan D., Folkers G., 3D QSAR Study of N2-phenylguanines as Inhibitors of Herpes Simplex Virus Thymide Kinase, Antiviral and Antitumoral Research, http://www.pharma.ethz.ch/text/research/tk/qsar.html.

[17] Kellogg G. E., Semus S. F., Abraham D. J., HINT: a new method of empirical hydrophobic field calculation for CoMFA, J. Comput.-Aided Mol. Des., 1991, 5(6), p. 545-552.

[18] Myers A. M., Charifson P. S., Owens C. E., Kula N. S., McPhail A. T., Baldessarini R. J., Booth R. G., Wyrick S. D., Conformational Analysis, Pharmacophore Identification, and Comparative Molecular Field Analysis of Ligands for the Neuromodulatory .sigma.3 Receptor, J. Med. Chem., 1994, 37(24), p. 4109-4117.

[19] Kim K. H., Use of the hydrogen-bond potential function in comparative molecular field analysis (CoMFA): An extension of CoMFA.

[20] Durst G. L., Comparative Molecular Field Analysis (CoMFA) of Herbicidal Protoporphyrinogen Oxidase Inhibitors using Standard Steric and Electrostatic Fields and an Alternative LUMO Field.

[21] Waller C.L., Marshall G. R., Three-Dimensional Quantitative Structure-Activity Relationship of Angiotensin-Converting Enzyme and Thermolysin Inhibitors. II. A Comparision of CoMFA Models Incorporating Molecular Orbital Fields and Desolvation Free Energy Based on Active-Analog and Complementary-Receptor-Field Alignment Rules, J. Med. Chem., 1993, 36, p. 2390-2403.

[22] Wiese M., Pajeva I. L., A Comparative Molecular Field Analysis of Propafenone-type Modulators of Cancer Multidrug Resistance, Quant. Struct.-Act. Relat., 1998, 17(4), p. 301-312.

[23] Klebe G., Abraham U., On the Prediction of Binding Properties of Drug Molecules by Comparative Molecular Field Analysis, J. Med. Chem., 1993, 36(1), p. 70-80.

[24] Czaplinski K.H.A., Grunewald G. L., A Comparative Molecular Field Analysis Derived Model of Binding of Taxol Analogs to Microtubes, Bioorg. Med. Chem. Lett., 1994, 4(18), p. 2211-2216.

[25] Akagi T., Exhaustive Conformational Searches for Superimposition and Three-Dimensional Drug Design of Pyrethroids, QSAR, 1998, 17(6), p. 565-570.

[26] Waller C.L., Oprea T.I., Giolitti A., Marshall G.R., Three-Dimensional QSAR of Human Immunodeficiency Virus. (I) Protease Inhibitors. 1. A determined Alignment Rules, J. Med. Chem., 1993, 36(26), p. 4152-4160.

[27] Thompson E., The Use of Substructure Search and Relational Databases for Examining the Carcinogenic Potential of Chemicals; Conference on Computational Methods in Toxicology – April, 1998, Holiday Inn/I-675, Dayton, Ohio, USA; abstract available at http://www.ccl.net /ccl/toxicology/abstracts/tabs6.html.

[28] Todeschini R., Lasagni M., Marengo E., New Molecular Descriptors for 2D and 3D Structures. Theory J. Chemometrics, 1994, 8, p. 263-272.

[29] Todeschini R., Gramatica P., Provenzani R., Marengo E., Weighted Holistic Invariant Molecular (WHIM) descriptors. Part2. There Development and Application on Modeling Physico-chemical Properties of Polyaromatic Hydrocarbons, Chemometrics and Intelligent Laboratory Systems, 1995, 27, p. 221-229.

[30] Todeschini R., Vighi M., Provenzani R., Finizio A., Gramatica P., Modeling and Prediction by Using WHIM Descriptors in QSAR Studies: Toxicity of Heterogeneous Chemicals on Daphnia Magna, Chemosphere, 1996, 8, p. 1527.

[31] Zaliani A., Gancia E., MS-WHIM Scores for Amino Acids: A New 3D-Description for Peptide QSAR and QSPR Studies, J. Chem. Inf. Comput. Sci., 1999, 39(3), p. 525-533.

[32] Bravi G., Gancia E., Mascagni P., Pegna M., Todeschini R., Zaliani A., MS-WHIM., New 3D Theoretical Descriptors Derived from Molecular Surface Properties: A Comparative 3D QSAR Study in a Series of Steroids, J. Comput.-Aided Mol. Des., 1997, 11, p. 79-92.

[33] Niemelä J.R., Validation of the BIODEG Probability Program, TemaNord Repost, 1994, 589, p. 153-156.

[34] Damborsky J., A mechanistic approach to deriving quantitative structure-activity relationship models for microbial degradation of organic compounds. SAR and QSAR in Environmental Research, Proceedings of the Satellite Workshop on Biodegradation accompanying the 6th International Workshop on QSAR in Environmental Sciences, Italy, September 12, 1994.

[35] Blok J., Classification of biodegradability by growth kinetic parameters, Ecotoxicology and Environmental Safety, 1994, 27, p. 294-305.

[36] Warne M. A., Ebbels T. M. D., Lindon J. C., Nicholson J. K., Semiempirical Molecular-Orbital Properties of Some Polycyclic Aromatic Hydrocarbons and Correlation with Environmental Toxic Equivalency Factors, Polycyclic Aromatic Compounds, 2003, 23, 23-74.

[37] Schultz T. W., Cronin M. T. D., Walker J. D., Aptula A. O., Quantitative structure- activity relationships (QSARs) in toxicology: a historical perspective, Journal of Molecular Structure: THEOCHEM, 2003, 622(1), p. 1-22.

[38] Schultz T. W., Cronin M. T. D., Netzeva T. I., The present status of QSAR in toxicology, Journal of Molecular Structure: THEOCHEM, 2003, 622, p. 23-38.

[39] Sparks T. C., Crouse G. D., Durst G., Natural products as insecticides: the biology, biochemistry and quantitative structure–activity relationships of spinosyns and spinosoids, Pest Management Science, 2001, 10, 896-905.

[40] Kompare B. Estimating environmental pollution by xenobiotic chemicals using QSAR (QSBR) models based on artificial intelligence, Water Science and Technology, 1998, 37(8), p. 9-18.

[41] Marjan Vracko, Kohonen Artificial Neural Network and Counter Propagation Neural Network in Molecular Structure-Toxicity Studies, Current Computer - Aided Drug Design, 2005, 1(1), p. 73-78.

[42] Sorana D. BOLBOACĂ, Lorentz JÄNTSCHI, Modelling the Inhibitory Activity on Carbonic Anhydrase I of Some Substituted Thiadiazoleand Thiadiazoline-Disulfonamides: Integration of Structure Information, ESCAPE17 - 17th European Symposium on Computer Aided Process Engineering, CAPE Working Party of the European Federation of Chemical Engineering then Elsevier Netherlands & UK, ISBN 978-0-444-53157-5 & eISBN 0-444-53158-2, Bucharest, Romania then Amsterdam, Netherlands, T4-212 (oral presentation), May 27-30 then p. 965-970, 2007.

[43] Sorana D. BOLBOACĂ, Lorentz JÄNTSCHI, Data Mining on Structure-Activity/Property Relationships Models, ECCC11- The 11th Electronic Computational Chemistry Conference, online, Monmouth University, New Jersey, USA, N/A, www, Internet, paper #29, Presentation located here, April 2-30, 2007.

[44] Sorana Daniela BOLBOACĂ, Lorentz JÄNTSCHI, Modeling the Property of Compounds from Structure: Statistical Methods for Models Validation, Institute of General and Inorganic Chemistry, Bulgarian Academy of Sciences [http://sizemat.igic.bas.bg], FP6: EC-INCO-CT-2005-016414 Specific Support Action, Plovdiv, Bulgaria, p. 71, April 19-21, 2007.

[45] Carmen Elena STOENOIU, Sorana Daniela BOLBOACĂ, Lorentz JÄNTSCHI, Characterization of Marine Sediments Toxicity based on Structural Information, Institute of General and Inorganic Chemistry, Bulgarian Academy of Sciences [http://sizemat.igic.bas.bg],

FP6: EC-INCO-CT-2005-016414 Specific Support Action, Plovdiv, Bulgaria, p. 54, April 19-21, 2007.

[46] Lorentz JÄNTSCHI, Carmen Elena STOENOIU, Sorana Daniela BOLBOACĂ, Use of Graph Polynomials for Characterization of Alkanes Properties, International Symposium on Organic Chemistry, December 9-12, 2006, Avangard Prisma, ISBN 954-323-243-1 & 978-954-323-243-7, Sofia, Bulgaria, p. 87-88, December 9-12, 2006.

[47] Lorentz JÄNTSCHI, Mircea V. DIUDEA, Subgraphs by Pairs of Vertices, TOPMOL2006 - 20 Years Anniversary of Molecular Topology at Cluj, Babes-Bolyai University, CEEx M3 PR-D11-PT00-41, Cluj-Napoca, Romania, p. 16 [Plenary oral presentation], September 25-30, 2006.

[48] Lorentz JÄNTSCHI, Sorana Daniela BOLBOACĂ, Molecular Descriptors Family on Structure-Activity and Structure-Property Relationships: Results, SizeMat: Workshop on Size-Dependent Effects in Materials for Environmental Protection and Energy Application, Institute of General and Inorganic Chemistry, Bulgarian Academy of Sciences [http://sizemat.igic.bas.bg], FP6: EC-INCO-CT-2005-016414 Specific Support Action, Varna, Bulgaria, p. 14-15 [Plenary oral presentation], May 25-27, 2006.

[49] Sorana Daniela BOLBOACĂ, Lorentz JÄNTSCHI, Activity Characterization of Triazines Analogues: Statistical Parameters for Models Assessment, International Symposium on Organic Chemistry, December 9-12, 2006, Avangard Prisma, ISBN 954-323-243-1 & 978-954-323-243-7, Sofia, Bulgaria, p. 48-49 [Plenary oral presentation], December 9-12, 2006.

[50] Lorentz JÄNTSCHI, Sorana-Daniela BOLBOACĂ, Online System for Molecular Descriptors Family on Structure-Activity Relationships: Assessment and Characterization of Biologic Active Compounds, 6th European Conference on Computational Chemistry, European Association for Chemical and Molecular Sciences, Tale, Slovakia.

[51] Ştefan ŢIGAN, Lorentz JÄNTSCHI, Sorana-Daniela BOLBOACĂ, Modeling Herbicidal Activity of a Substituted Triazines Class by Integration of Compounds Complex Structural Information, XXIII International Biometric Conference, International Biometric Society, Montreal, Quebec, Canada, TP1.219 (509.pdf on CD), July 16-21, 2006.

[52] Lorentz JÄNTSCHI, Sorana-Daniela BOLBOACĂ, Modeling the Octanol-Water Partition Coefficient of Substituted Phenols: the Use of Structure Information, Third Humboldt Conference on Computational Chemistry, InnosLab Ltd., ISBN 954-323-199-0 & 978-954-323-199-7, Varna, Bulgaria, p. 65, June 24-28, 2006.

[53] Sorana Daniela BOLBOACĂ, Ştefan ŢIGAN, Lorentz JÄNTSCHI, Molecular Descriptors Family on Structure-Activity Relationships on anti-HIV-1 Potencies of HEPTA and TIBO Derivatives, Integrating Biomedical Information: From eCell to ePatient, European Federation

for Medical Informatics, ISBN 3-89838-0722-6 (Aka, Germany), ISBN 1-58603-614-9 (IOS Press, Holland), ISBN 973-625-303-1 (Politehnica Timişoara, Romania), Timisoara, Romania, p. 110-114, April 6-8, 2006.

[54] Lorentz JÄNTSCHI, Sorana BOLBOACĂ, Modelling the Inhibitory Activity on Carbonic Anhydrase IV of Substituted Thiadiazole- and Thiadiazoline- Disulfonamides: Integration of Structure Information, New Frontiers in Medicinal Chemistry, 1st European Chemistry Congress, 2006 August 27-31, Budapest, Hungary, abstract.

[55] Lorentz JÄNTSCHI, Sorana BOLBOACĂ, Pearson versus Spearman, Kendall's Tau Correlation Analysis on Structure-Activity Relationships of Biologic Active Compounds, ISCB27 - International Society for Clinical Biostatistics, 2006 August 27-31, Geneva, Switzerland, Conference Program, abstract.

[56] Sorana Daniela BOLBOACĂ, Lorentz JÄNTSCHI, Structure-Based Prediction of Alkanes Boiling Points, 2nd Workshop on Computational Chemistry and Its Applications (2nd CCA), 2007 May 27-30, Beijing, China.

[57] Sorana D. BOLBOACĂ, Lorentz JÄNTSCHI, From Structural Chemistry to Structural Biology through Information Science Study on Colagen, Math/Chem/Comp 2007 - The 22nd International Course & Conference on the Interfaces among Mathematics, Chemistry & Computer Sciences, Dubrovnik, Croatia, 2007 June 11 - 16.

[58] Sorana D. BOLBOACĂ, Lorentz JÄNTSCHI, Growth Inhibition Activity Analysis of Taxoids: Integration of Structure Information, Math/Chem/Comp 2007 - The 22nd International Course & Conference on the Interfaces among Mathematics, Chemistry & Computer Sciences, Dubrovnik, Croatia, 2007 June 11 - 16.

[59] Lorentz JÄNTSCHI, Carmen Elena STOENOIU and Sorana-Daniela BOLBOACĂ, A Formula for Vertex Cuts in b-Trees, Fourth International Conference of Applied Mathematics and Computing, Plovdiv, Bulgaria, 2007 August 12 - 18.

[60] Lorentz JÄNTSCHI, Sorana-Daniela BOLBOACĂ, and Cristina Maria FURDUI, Characteristic and Counting Polynomials on Modeling Nonane Isomers Properties, Fourth International Conference of Applied Mathematics and Computing, Plovdiv, Bulgaria, 2007 August 12 - 18.

[61] Sorana-Daniela BOLBOACĂ and Lorentz JÄNTSCHI, Structure-Property Based Model Estimation of Alkanes Boiling Points, Fourth International Conference of Applied Mathematics and Computing, Plovdiv, Bulgaria, 2007 August 12 - 18.

[62] Lorentz JÄNTSCHI, Sorana D. BOLBOACĂ, Antiallergic Activity of Substituted Benzamides: Characterization, Estimation and Prediction, Clujul Medical, Editura Iuliu Hatieganu, ISSN 1222-2119, Cluj-Napoca, Romania, LXXX(1), p. 125-132, 2007.

[63] Sorana D. BOLBOACĂ, Lorentz JÄNTSCHI, Modelling the Inhibitory Activity on Carbonic Anhydrase I of Some Substituted Thiadiazoleand Thiadiazoline-Disulfonamides: Integration of Structure Information, Computer-Aided Chemical Engineering, Elsevier Netherlands & UK, eISSN 1570-7946, Amsterdam, Netherlands, 24(2007), p. 965-970, 2007.

[64] Lorentz JÄNTSCHI, Sorana D. BOLBOACĂ, The Jungle of Linear Regression Revisited, Leonardo Electronic Journal of Practices and Technologies, AcademicDirect, ISSN 1583-1078, www, Internet, 6(10), p. 169-187, 2007.

[65] Sorana D. BOLBOACĂ, Lorentz JÄNTSCHI, How Good the Characteristic Polynomial Can Be for Correlations?, International Journal of Molecular Sciences, Molecular Diversity Preservation International, ISSN 1422-0067, www & Basel, Internet & Switzerland, 8(4), p. 335-345, 2007, 2007.

[66] Lorentz JÄNTSCHI, Sorana-Daniela BOLBOACĂ, Modeling the Octanol-Water Partition Coefficient of Substituted Phenols by the Use of Structure Information, International Journal of Quantum Chemistry, Wiley InterScience, ISSN 0020-7608, eISSN 1097-461X, New York, NY, USA, Volume 107, Issue 8 (Special Issue: Proceedings from the 3rd Humboldt Conference on Computational Chemistry. Issue Edited by Georgi N. Vayssilov, Tzonka Mineva), p. 1736-1744, 2007.

[67] Lorentz JÄNTSCHI, Sorana BOLBOACĂ, Results from the Use of Molecular Descriptors Family on Structure Property/Activity Relationships, International Journal of Molecular Sciences, Molecular Diversity Preservation International, ISSN 1422-0067, www & Basel, Internet & Switzerland, 8(3), p. 189-203, 2007.

[68] Sorana D. BOLBOACĂ, Lorentz JÄNTSCHI, Molecular Descriptors Family on Structure-Activity Relationships: Modeling Herbicidal Activity of Substituted Triazines Class, Bulletin of University of Agricultural Sciences and Veterinary Medicine - Agriculture, AcademicPres, ISSN 1454-2382, Cluj-Napoca, Romania, 62, p. 35-40, 2006.

[69] Lorentz JÄNTSCHI, Sorana BOLBOACĂ, Modelling the Inhibitory Activity on Carbonic Anhydrase IV of Substituted Thiadiazole- and Thiadiazoline- Disulfonamides: Integration of Structure Information, Electronic Journal of Biomedicine, Red UniNet Spain, ISSN 1697-090X, www, Internet, 2006(2), p. 22-33, 2006.

[70] Sorana BOLBOACĂ, Lorentz JÄNTSCHI, Pearson Versus Spearman, Kendall's Tau Correlation Analysis on Structure-Activity Relationships of Biologic Active Compounds, Leonardo Journal of Sciences, AcademicDirect, ISSN 1583-0233, www, Internet, 5(9), p. 179-200, 2006.

[71] Sorana BOLBOACĂ, Claudia FILIP, Ştefan ŢIGAN, Lorentz JÄNTSCHI, Antioxidant Efficacy of 3-Indolyl Derivates by Complex Information Integration, Clujul Medical, Editura Iuliu Hatieganu, ISSN 1222-2119, Cluj-Napoca, Romania, LXXIX(2), p. 204-209, 2006.

[72] Sorana Daniela BOLBOACĂ, Lorentz JÄNTSCHI, Modeling of Structure-Toxicity Relationship of Alkyl Metal Compounds by Integration of Complex Structural Information, Therapeutics: Pharmacology and Clinical Toxicology, RP Press, ISSN 1583-0012, Bucuresti, Romania, X(1), p. 110-114, 2006.

[73] Lorentz JÄNTSCHI, Sorana BOLBOACĂ, Molecular Descriptors Family on Structure Activity Relationships 6. Octanol-Water Partition Coefficient of Polychlorinated Biphenyls, Leonardo Electronic Journal of Practices and Technologies, AcademicDirect, ISSN 1583-1078, www, Internet, 5(8), p. 71-86, 2006.

[74] Lorentz JÄNTSCHI, Sorana BOLBOACĂ, Molecular Descriptors Family on Structure Activity Relationships 5. Antimalarial Activity of 2,4-Diamino-6-Quinazoline Sulfonamide Derivates, Leonardo Journal of Sciences, AcademicDirect, ISSN 1583-0233, www, Internet, 5(8), p. 77-88, 2006.

[75] Lorentz JÄNTSCHI, Mihaela Ligia UNGUREŞAN, Sorana Daniela BOLBOACĂ, Complex Structural Information Integration: Inhibitor Activity on Carbonic Anhydrase II of Substituted Disulfonamides, Applied Medical Informatics, SRIMA, ISSN 1224-5593, Cluj-Napoca, Romania, 17(3-4), p. 12-21, 2005.

[76] Sorana Daniela BOLBOACĂ, Lorentz JÄNTSCHI, Structure-Property Based Model for Alkanes Boiling Points, International Journal of Pure and Applied Mathematics, July 25, 2007.

[77] Sorana D. BOLBOACĂ and Lorentz JÄNTSCHI, Modelling the Property of Compounds from Structure: Statistical Methods for Models Validation, Environmental Chemistry Letters.

[78] Carmen E. STOENOIU, Sorana D. BOLBOACĂ and Lorentz JÄNTSCHI, Characterization of Marine Sediments Toxicity Based on Structure Information, Environmental Chemistry Letters.

[79] Sorana D. BOLBOACĂ and Lorentz JÄNTSCHI, From Structural Chemistry to Structural Biology through Information Science Study on Collagen, Chemical Biology & Drug Design.

[80] Sorana D. BOLBOACĂ, Lorentz JÄNTSCHI, Molecular Descriptors Family on Structure-Activity Relationship Assessment, Environmental Toxicology & Chemistry.

[81] Sorana Daniela BOLBOACĂ, Lorentz JÄNTSCHI, Data Mining on Structure-Activity/Property Relationships Models, World Applied Sciences Journal.

[82] Sorana Daniela BOLBOACĂ, Lorentz JÄNTSCHI, Structure versus Biological Role of Substituted Thiadiazole- and Thiadiazoline- Disulfonamides, Studii si Cercetari de Biologie, Seria biologie vegetala.

[83] Lorentz JÄNTSCHI, Sorana Daniela BOLBOACĂ, Triazines Herbicidal Assessed Activity, Studii si Cercetari de Biologie, Seria biologie vegetala.

[84] Lorentz JÄNTSCHI, Book review: Periodic Nanostructures, by Diudea M.V. and Nagy C.L., ISBN 978-1-4020-6019-9, Springer, June 2007, Studii si Cercetari de Biologie, Seria biologie vegetala.

[85] Lorentz JÄNTSCHI and Sorana D. BOLBOACĂ, Counting Polynomials on Regular Iterative Structures, Journal of Graph Theory, June 7, 2007.

[86] Lorentz JÄNTSCHI, Mircea V. DIUDEA, Subgraphs of Pair Vertices, Journal of Mathematical Chemistry (from autumn 2006).

[87] Lorentz JÄNTSCHI and Sorana D. BOLBOACĂ, Molecular Descriptors Family on Chromatography, BBCAC-4 4th Black Sea Basin Conference on Analytical Chemistry, Sunny Beach, Bulgaria, 2007 September 19-23.

[88] Lorentz JÄNTSCHI, Characteristic and Counting Polynomials of Nonane Isomers, AcademicDirect, , ISBN 973-86211-3-5 & ISBN 980-973-86211-3-8, www, Internet, p. 101, 2007.

[89] Lorentz JÄNTSCHI, Mădălina Ana VĂLEANU, Sorana Daniela BOLBOACĂ, Rapid Programming of Relational Databases Applications (in Romanian), AcademicDirect & AcademicPres, , ISBN: 973-86211-5-1 & 978-973-86211-5-2 (AcademicDirect) && ISBN 973-744-044-7 & 978-973-744-044-0 (AcademicPres), www & Cluj-Napoca, Internet & Romania, 233 p., 2006.

[90] Horea Iustin NAŞCU, Lorentz JÄNTSCHI, Instrumental and Analytical Chemistry (in Romanian), AcademicDirect & AcademicPres, , ISBN 973-86211-4-3 & 978-973-86211-4-5 (AcademicDirect) && ISBN 973-744-046-3 & 978-973-744-046-4 (AcademicPres), www & Cluj-Napoca, Internet & Romania, 320 p., 2006.

[91] Monica ŞTEFU, Daniela BUTYKA, Mircea V. DIUDEA, Lorentz JÄNTSCHI, Bazil PÂRV, Nanostructures: Novel Architecture - Mircea V. DIUDEA (ed.), Chapter: Algorithms for Basic Operations on Maps, Nova Science, ISBN 1-59454-499-9, Huntington, New York, USA, 420 p., 98$, 2005.

[92] Sorana D. BOLBOACĂ and Lorentz JÄNTSCHI, From Molecular Structure to Molecular Design through Molecular Descriptors Family Methodology, in Eduardo A. CASTRO, QSPR-QSAR studies on desired properties for drug design, December 2007.

Concluzii

Implementarea sistemului online pentru investigații structurale complexe pe compuși chimici biologic activi a fost realizată și și-a dovedit acuratețea estimativă (prin intermediul coeficienților de corelație și determinare obținuți pe seturile investigate, constant superioare altor metode) și predictivă (prin intermediul investigațiilor folosind metodele leave-one-out și traning-vs-test). Analiza corelaților corelate, aplicată în majoritatea cazurilor studiate, a arătat de asemenea că modelele structură-activitate obținute au fost semnificativ mai bune decât modelele anterioare.

Se impune realizarea a încă unui pas, și anume aplicarea metodei în domeniul drug design, având ca suport chimia combinatorială cu implicații majore în planul sănătății, mediului și alimentației.