Extracting the Common Structure of Compounds to Induce Plant Immunity Activation using ILP

Atsushi matsumoto,[1, 2] Katsutoshi Kanamori,[1] Kazuyuki Kuchitsu,[3]
and Hayato Ohwada[1]

1. Department of Industrial Administration, Faculty of Science and Technology,
Tokyo University of Science
2. 7415617@ed.tus.ac.jp
3. Department of Applied Biological Science, Faculty of Science and Technology,
Tokyo University of Science

Abstract. While recent studies have referred to plant immunity activators, it is difficult to find a compound to use for the immunity activation of plants. In this study, we seek to determine compounds that enable plant immunity activity using ILP. With the proposed method, it is possible to predict compounds that induce plant immunity activity, based on the structural features of the compounds. The predicted structure rule also includes structures of known plant immunity activators. However, further investigation is needed regarding the relationship between plant immunity and structure rules.

Keywords: ILP, Machine learning, Plant immunity activation, Virtual screening

1. INTRODUCTION

Virtual screening is an important approach in the drug discovery process. Especially, machine learning has recently received broad attention. This paper picks up two method, Support Vector Machine (SVM) [1] and Inductive Logic Programming (ILP). Both method are often used in drug discovery field [2], [3].

On the other hand, decreased production of agricultural crops due to pathogenic bacteria and pests is a serious problem that has not yet been solved. To address this problem, grower have made a deal with fungicides and pesticides, however, it is difficult to act selectively on the target (e.g., pests and pathogens). There is a possibility that the cause of health damage in humans and destruction of biota. In addition, long-term use of the same drug may cause the emergence of resistant bacteria; thus, the effect of the drug gradually decreases. In recent years, plant immunity activators have attracted attention, based on the idea of increasing the immunity of the plant rather than directly killing pathogens and pests. However, only three types of plant immunity activator are currently marketed in Japan (Fig. 1). In addition, the mechanism of plant-immunity activation is still largely unknown [4].

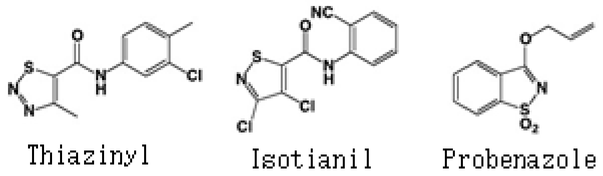


Fig. 1. Known plant-immunity activators

The development of plant immunity activators has been slow, due to the time required and the high cost of screening candidate compounds. Cause of this problem is the kind of candidate compounds is enormous and each of the compounds were reacted to the cells to confirm the effect of immunity activation.

In this study, we predict compounds that induce plant-immunity activation using ILP to study compound structures. ILP can be used to determine relationship patterns between data; therefore, it is suitable to represent the structure of compounds. Additionally, we obtained the structure of the predicted compound as a rule, which is one of the excellent points of ILP. A recent study that was conducted to predict the structure of compounds using ILP exhibited high performance [5]. In those cases, the target of compound bonds was known. However, in the present study, the target of compound bonds is not known. Additionally, we also tried SVM for comparison with ILP. SVM also exhibited high performance [2].

2. PLANT IMMUNITY

Plant immunity is a defense system to protect plants from various enemies. A plant-immunity activator is a drug that activates plant immunity. The Kuchitu group constructed a screening system to find a candidate using the amount of ROS (reactive oxygen species) generation as an index [6]. Experiment results indicated that if the ROS value is high, the compound is likely to be a plant-immunity activator.

3. DATASET

In the present study, the datasets are experiment data about the plant immunity activator in Arabidopsis thaliana, compiled by the Kuchitu group. This dataset includes 10000 compounds. Positive examples are 271 high-ROS compounds, and negative examples are the other 9729 compounds. However, negative examples were reduced to 813 compounds by random sampling for two reasons. First, imbalanced data deteriorates learning accuracy. Second, if there are many compounds, calculation takes a long time. Therefore, 1084 compounds were used in this study.

4. METHOD

This chapter describes our method. We had two approaches. Fig. 2 shows the overview of our method.

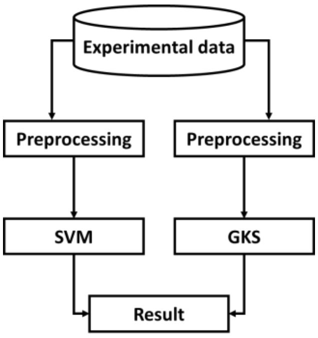


Fig. 2. Method overview

The two approaches are described as follow.

4.1 ILP Approach

With the ILP approach, structural features and some numerical features of the compound were used as background knowledge. In this study, we used GKS [7], which is an ILP system. We defined seven predicates to represent the features of the compounds. In parentheses, there are argument of predicates.

・atom (compound_name, atom_id, element)
Types of atoms present in the compound

・bond (compound_name, atom_id, atom_id, bondtype)
Bonding state between atoms and bond type in the compound

・Num_AromaticRings (compound_name,Num_AromaticRing)
The number of aromatic rings in the compound

・Num_Rings (compound_name, Num_Ring)
The number of rings in the compound

・LogP98 (compound_name, value)
Lipid solubility of the compound

・LogD (compound_name, value)
Indication of a change in lipid solubility by a change in Ph value

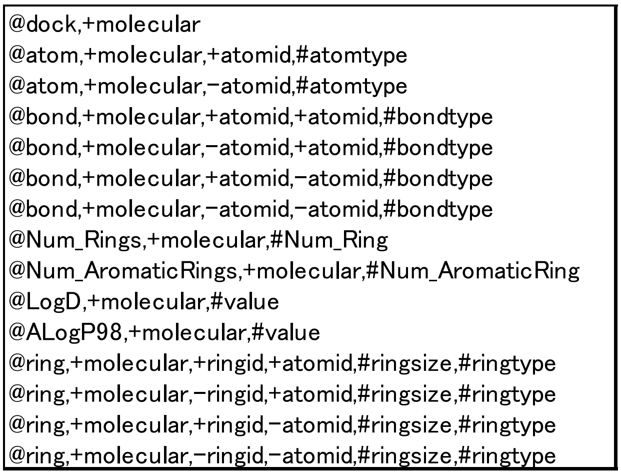
- ring (compound_name,ring_id,atom_id,ringsize,ringtype)
  Type of ring structure that is composed of each atom. It can represent the connection of the ring structure and other structures by using this predicate.

By selecting several predicates as background knowledge, we can obtain the structure of the compound as a learning result (Table 1). Background knowledge is a set of atomic formulas of each predicate. *Atom* and *bond* are always necessary. The reason why selecting *LogP98* and *LogD* is result of importance calculation using the average Gini coefficient.

Table 1. Predicates selected for background knowledge

Setting name	Predicate
ILP1	atom,bond
ILP2	atom,bond,Num_AromaticRings
ILP3	atom,bond,Num_AromaticRings,Num_rings
ILP4	atom,bond,ALogP98
ILP5	atom,bond,Num_AromaticRings,Num_rings,ALogP98,LogD
ILP6	atom,bond,Num_AromaticRings,Num_rings,LogD
ILP7	atom,bond,LogD,ring
ILP8	atom,bond,ring

Mode declaration as input is shown in Fig. 3. A rule selected if it was covered more than 10 positive examples and less than 10 negative examples.



```
@dock,+molecular
@atom,+molecular,+atomid,#atomtype
@atom,+molecular,−atomid,#atomtype
@bond,+molecular,+atomid,+atomid,#bondtype
@bond,+molecular,−atomid,+atomid,#bondtype
@bond,+molecular,+atomid,−atomid,#bondtype
@bond,+molecular,−atomid,−atomid,#bondtype
@Num_Rings,+molecular,#Num_Ring
@Num_AromaticRings,+molecular,#Num_AromaticRing
@LogD,+molecular,#value
@ALogP98,+molecular,#value
@ring,+molecular,+ringid,+atomid,#ringsize,#ringtype
@ring,+molecular,−ringid,+atomid,#ringsize,#ringtype
@ring,+molecular,+ringid,−atomid,#ringsize,#ringtype
@ring,+molecular,−ringid,−atomid,#ringsize,#ringtype
```

Fig. 3. Mode declaration

4.2 SVM Approach

We also tried SVM for comparison with ILP, using 77 features for learning (Table 2). Detail information is shown in Appendix A.

Table 2. Attributes used for SVM

Types of features	The number of features
Related to structure	39
Related to ALogP	6
Related to size or weight	14
Related to energy	12
Other	6
Total	77

Cost parameters and gamma parameters were determined using a grid search for 20 split from 0.0001 to 10,000. The kernel used RBF.

4.3 Evaluation

Ten-fold cross-validation was used in both approaches. True Positive (tp) , False Negative (fn) , True Negative (tn) , False Positive (fp) , Accuracy , Precision , Recall and F value were used for Evaluation. Especially, this paper focuses on tp and F value.

5. RESULTS

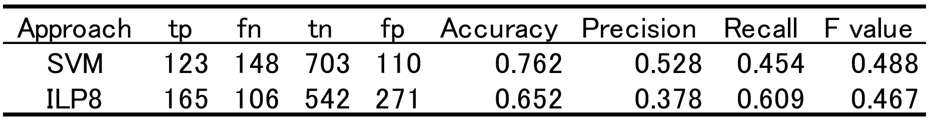
Table 3 shows the ILP results.

Table 3. ILP results

Setting name	tp	fn	tn	fp	Accuracy	Precision	Recall	F value
ILP1	92	179	699	114	0.73	0.447	0.339	0.386
ILP2	116	155	644	169	0.701	0.407	0.428	0.417
ILP3	127	144	605	208	0.675	0.379	0.469	0.419
ILP4	88	183	712	101	0.738	0.466	0.325	0.383
ILP5	131	140	572	241	0.649	0.352	0.483	0.407
ILP6	139	132	568	245	0.652	0.362	0.513	0.424
ILP7	165	106	523	290	0.635	0.363	0.609	0.455
ILP8	165	106	542	271	0.652	0.378	0.609	0.467

Table 4 shows comparison of the best of SVM and the best of ILP

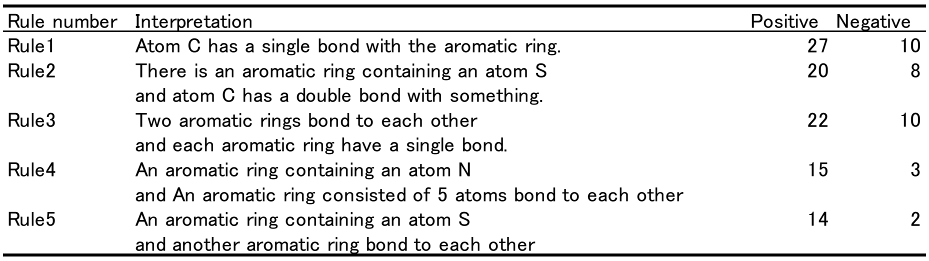
Table 4. Comparison of the best of SVM and the best of ILP



Approach	tp	fn	tn	fp	Accuracy	Precision	Recall	F value
SVM	123	148	703	110	0.762	0.528	0.454	0.488
ILP8	165	106	542	271	0.652	0.378	0.609	0.467

Table 5 shows the best rules obtained by ILP8. A good rule has many positive examples and few negative examples. All the output list of rules obtained by ILP8 are shown in Appendix B

Table 5. Rules for compound structure



Rule number	Interpretation	Positive	Negative
Rule1	Atom C has a single bond with the aromatic ring.	27	10
Rule2	There is an aromatic ring containing an atom S and atom C has a double bond with something.	20	8
Rule3	Two aromatic rings bond to each other and each aromatic ring have a single bond.	22	10
Rule4	An aromatic ring containing an atom N and An aromatic ring consisted of 5 atoms bond to each other	15	3
Rule5	An aromatic ring containing an atom S and another aromatic ring bond to each other	14	2

6. CONCLUSION

Although SVM F values slightly exceeded those of ILP, ILP tp values greatly exceeded those of SVM. For virtual screening, it is very important to reduce the positive example of misclassification. Results of this study indicate that structural features of the compounds are useful in predicting immunity activation.

Using the ring structure as background knowledge yielded better results than not using ring structure. Therefore, the ring structure is considered an important factor in plant immunity activation.

When analyzing rules using ILP, comparison of known plant immunity activators indicated that Rule 2 was true for all three compounds. For rule showing a structure that is different from the known plant immunity activator, there is a need for further investigation.

In this study, it was possible to predict the partial structure that exists in all compounds of known plant-immunity activators. In addition, the rule that is unknown the relationship between immunity activity has been predicted. In order to improve prediction accuracy, it is essential to improve background knowledge in the future.

References

1. V.Vapnik，The Nature of Statistical learning Theory. Spring-Verlag，NY，USA，1995
2. Tadasuke Ito，Hayato Ohwada and Shin Aoki，Combining two machine learning methods for predicting protein-ligand docking using structure and physiochemical properties，Proc. of the 7th International Conference on Bioinformatics and Computational Biology， pp. 19-24，March 2015
3. A.Srinivasan，S.H.Muggleton，R.D.King and M.J.E.Sternberg，Mutagenesis ILP experiments in a non-determinate biological domain，Proceedings of the Fourth International Inductive Logic Programming Workshop，1994
4. Yoshiteru Noutoshi，Masateru Okazaki，Tatsuya Kida，Yuta Nishina，Yoshihiko Morishita，Takumi Ogawa，Hideyuki Suzuki，Daisuke Shibata，Yusuke Jikumaru，Atsushi Hamada，Yuji Kamiya，Ken Shirasu，Novel Plant Immune-Priming Compounds Identified via High-Throughput Chemical Screening Target Salicylic Acid Glucosyltransferases in Arabidopsis．The Plant Cell，vol.24:3795-3804，2012
5. Jose C A Santos，Houssam Nassif，David Page，Stephen H Muggleton，Michael J E Sternberg，Automated identification of protein-ligand interaction features using Inductive Logic Programming:a hexose binding case study．Santos st al．BMC Bioinformatics 2012，13:162，2012
6. T Higashi，T Kurusu，S Hasegawa，K Kuchitsu，Dynamic intracellular reorganization of cytoskeletons and the vacuole in defense responses and hypersensitive cell death in plants．Journal of Plant Research，Volume 124，Issue 3，pp315-324，2011
7. Hayato Ohwada，Hiroyuki Nishiyama，Fumio Mizoguchi，Concurrent execution of optimal hypoyhesis search for inverse entailment．Lecture Notes in Artificial Intelligence，Spring-Verlag，No.1866，Vol.4，pp.165-173，2000

Appendix: A

Table 6 shows feature list in SVM approach. Feature name depends on Discovery Studio.

Table 6. Feature list in SVM

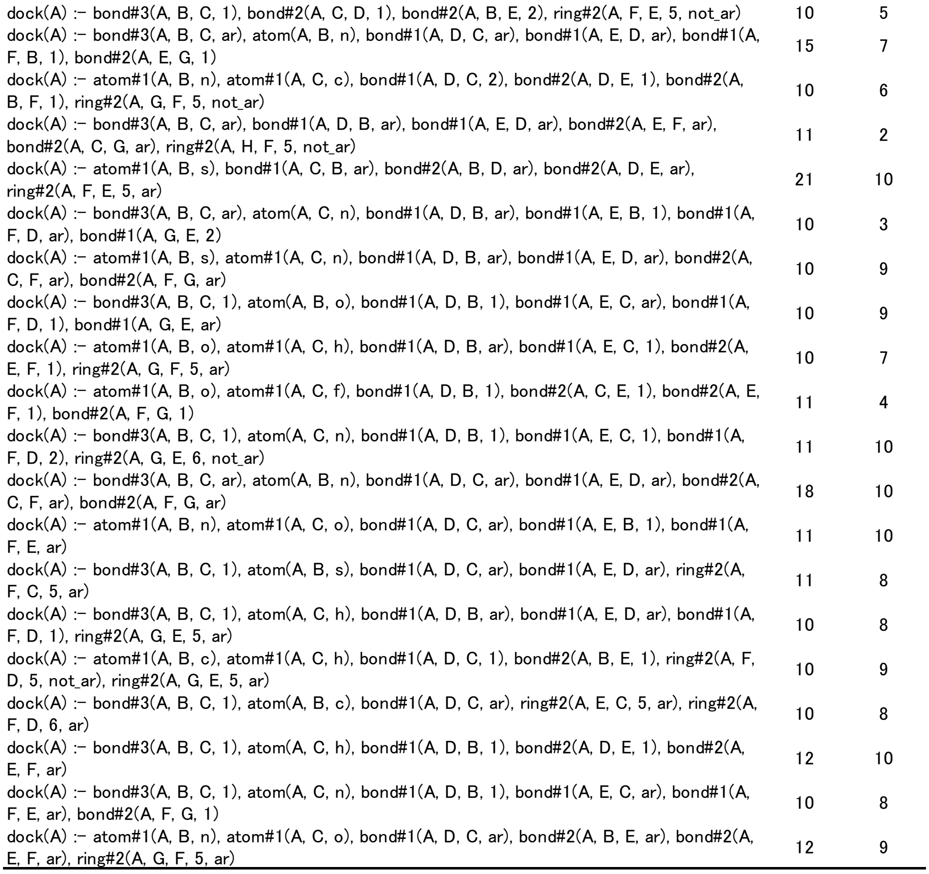
Category	Feature name	Category	Feature name
C	HBA_Count	A	ALogP
	HBD_Count		ALogP_MR
	NPlusO_Count		ALogP98
	Num_AromaticBonds		ALogP98_Unknown
	Num_AromaticRings		Apol
	Num_AtomClasses		LogD
	Num_Atoms	W	Molecular_3D_PolarSASA
	Num_Bonds		Molecular_3D_SASA
	Num_BridgeBonds		Molecular_3D_SAVol
	Num_BridgeHeadAtoms		Molecular_FractionalPolarSASA
	Num_ChainAssemblies		Molecular_FractionalPolarSurfaceArea
	Num_Chains		Molecular_Mass
	Num_ExplicitAtoms		Molecular_PolarSASA
	Num_ExplicitBonds		Molecular_PolarSurfaceArea
	Num_ExplicitHydrogens		Molecular_SASA
	Num_H_Acceptors		Molecular_SAVol
	Num_H_Acceptors_Lipinski		Molecular_SurfaceArea
	Num_H_Donors		Molecular_Volume
	Num_H_Donors_Lipinski		Molecular_Weight
	Num_Hydrogens		VSA_TotalArea
	Num_NegativeAtoms	E	Angle Energy
	Num_PositiveAtoms		Bond Energy
	Num_RingAssemblies		CHARMm Energy
	Num_RingBonds		Dihedral Energy
	Num_Rings		Electrostatic Energy
	Num_Rings3		Energy
	Num_Rings5		Improper Energy
	Num_Rings6		Initial Potential Energy
	Num_Rings7		Minimized_Energy
	Num_Rings8		Potential Energy
	Num_RotatableBonds		Strain_Energy
	Num_SpiroAtoms		Van der Waals Energy
	Num_StereoAtoms	O	AverageBondLength
	Num_StereoBonds		FormalCharge
	Num_TerminalRotomers		Initial RMS Gradient
	Num_TrueStereoAtoms		Molecular_Solubility
	Num_UnknownPseudoStereoAtoms		RadOfGyration
	Num_UnknownTrueStereoAtoms		RMS Gradient
	Organic_Count		

C: Related to structure A: Related to AlogP W: Related to size or weight
E: Related to energy O: Other

Appendix: B

Fig.4 show all the output list of rules obtained by ILP8.

Rule	Positive	Negative
dock(A) :- atom#1(A, B, s), atom#1(A, C, c), bond#1(A, D, C, 2), bond#2(A, B, E, ar), bond#2(A, E, F, ar)	20	8
dock(A) :- bond#3(A, B, C, 2), bond#1(A, D, C, 1), bond#2(A, B, E, 1), bond#2(A, E, F, 1)	10	2
dock(A) :- bond#3(A, B, C, ar), bond#1(A, D, C, 1), bond#1(A, E, B, ar), bond#1(A, F, D, 3)	10	8
dock(A) :- bond#3(A, B, C, 1), bond#1(A, D, B, ar), bond#1(A, E, D, 1), ring#2(A, F, E, 6, ar)	17	6
dock(A) :- bond#3(A, B, C, ar), atom(A, B, n), bond#1(A, D, B, 1), bond#2(A, C, E, ar)	14	10
dock(A) :- bond#3(A, B, C, ar), atom(A, B, s), bond#1(A, D, C, 1), bond#1(A, E, D, ar)	14	2
dock(A) :- atom#1(A, B, c), bond#1(A, C, B, ar), bond#1(A, D, C, ar), bond#2(A, B, E, 1), ring#2(A, F, E, 6, ar)	27	10
dock(A) :- atom#1(A, B, n), atom#1(A, C, n), bond#1(A, D, B, 2), bond#1(A, E, C, 1), ring#2(A, F, D, 6, not_ar)	10	9
dock(A) :- bond#3(A, B, C, ar), atom(A, C, n), bond#1(A, D, B, 1), bond#1(A, E, B, ar), bond#1(A, F, D, 2), bond#2(A, E, G, 1)	11	3
dock(A) :- atom#1(A, B, n), atom#1(A, C, o), bond#1(A, D, C, 1), bond#1(A, E, D, 1), ring#2(A, F, B, 5, ar)	18	8
dock(A) :- atom#1(A, B, c), bond#1(A, C, B, 2), bond#1(A, D, B, 1), bond#1(A, E, D, ar), bond#2(A, C, F, 1)	11	8
dock(A) :- bond#3(A, B, C, ar), bond#1(A, D, B, ar), bond#1(A, E, C, 1), bond#2(A, D, F, 1), ring#2(A, G, E, 6, ar)	20	10
dock(A) :- atom#1(A, B, n), atom#1(A, C, c), bond#1(A, D, C, ar), bond#1(A, E, D, 1), bond#2(A, B, F, ar), ring#2(A, G, E, 5, not_ar)	10	9
dock(A) :- atom#1(A, B, n), atom#1(A, C, c), bond#1(A, D, B, ar), bond#1(A, E, D, 1), bond#2(A, C, F, ar), ring#2(A, G, F, 6, not_ar)	15	10
dock(A) :- bond#3(A, B, C, ar), atom(A, B, n), bond#1(A, D, C, ar), bond#2(A, C, E, ar), bond#2(A, E, F, ar), bond#2(A, D, G, 1)	20	10
dock(A) :- bond#3(A, B, C, 1), bond#1(A, D, B, 1), bond#2(A, D, E, 1), ring#2(A, F, E, 6, not_ar), ring#2(A, G, C, 5, ar)	10	5
dock(A) :- atom#1(A, B, n), bond#2(A, B, C, 2), bond#2(A, C, D, 1), ring#2(A, E, D, 5, not_ar)	10	8
dock(A) :- atom#1(A, B, n), atom#1(A, C, n), bond#2(A, C, D, ar), bond#2(A, D, E, ar), bond#2(A, E, F, ar), ring#2(A, G, B, 6, not_ar)	16	10
dock(A) :- atom#1(A, B, c), bond#1(A, C, B, ar), bond#1(A, D, C, 1), bond#1(A, E, D, ar), bond#2(A, B, F, 1), bond#2(A, E, G, 1)	22	10
dock(A) :- bond#3(A, B, C, 1), atom(A, B, c), bond#1(A, D, C, 2), bond#2(A, C, E, 1), ring#2(A, F, E, 5, ar)	15	10
dock(A) :- bond#3(A, B, C, ar), atom(A, B, n), bond#1(A, D, C, ar), bond#1(A, E, C, 1), bond#1(A, F, D, 1), bond#1(A, G, F, ar)	12	10
dock(A) :- atom#1(A, B, c), bond#1(A, C, B, 2), bond#2(A, B, D, 1), bond#2(A, D, E, ar), bond#2(A, C, F, 1)	11	10
dock(A) :- atom#1(A, B, n), atom#1(A, C, h), bond#1(A, D, B, 2), bond#1(A, E, D, 1), bond#1(A, F, C, 1), ring#2(A, G, F, 6, not_ar)	11	10
dock(A) :- atom#1(A, B, c), atom#1(A, C, o), bond#1(A, D, B, ar), bond#1(A, E, D, 1), bond#1(A, F, E, ar), bond#2(A, C, G, ar)	11	10
dock(A) :- atom#1(A, B, n), bond#2(A, B, C, 2), bond#2(A, B, D, 1)	12	10
dock(A) :- atom#1(A, B, n), atom#1(A, C, o), bond#2(A, B, D, ar), bond#2(A, D, E, ar), ring#2(A, F, C, 6, not_ar)	11	10
dock(A) :- atom#1(A, B, o), atom#1(A, C, n), bond#1(A, D, C, ar), bond#2(A, B, E, ar), bond#2(A, D, F, 1)	10	7
dock(A) :- bond#3(A, B, C, ar), atom(A, B, n), bond#2(A, C, D, ar), bond#2(A, D, E, ar), bond#2(A, E, F, 1), ring#2(A, G, F, 5, ar)	15	3



dock(A) :- bond#3(A, B, C, 1), bond#2(A, C, D, 1), bond#2(A, B, E, 2), ring#2(A, F, E, 5, not_ar)	10	5
dock(A) :- bond#3(A, B, C, ar), atom(A, B, n), bond#1(A, D, C, ar), bond#1(A, E, D, ar), bond#1(A, F, B, 1), bond#2(A, E, G, 1)	15	7
dock(A) :- atom#1(A, B, n), atom#1(A, C, c), bond#1(A, D, C, 2), bond#2(A, D, E, 1), bond#2(A, B, F, 1), ring#2(A, G, F, 5, not_ar)	10	6
dock(A) :- bond#3(A, B, C, ar), bond#1(A, D, B, ar), bond#1(A, E, D, ar), bond#2(A, E, F, ar), bond#2(A, C, G, ar), ring#2(A, H, F, 5, not_ar)	11	2
dock(A) :- atom#1(A, B, s), bond#1(A, C, B, ar), bond#2(A, B, D, ar), bond#2(A, D, E, ar), ring#2(A, F, E, 5, ar)	21	10
dock(A) :- bond#3(A, B, C, ar), atom(A, C, n), bond#1(A, D, B, ar), bond#1(A, E, B, 1), bond#1(A, F, D, ar), bond#1(A, G, E, 2)	10	3
dock(A) :- atom#1(A, B, s), atom#1(A, C, n), bond#1(A, D, B, ar), bond#1(A, E, D, ar), bond#2(A, C, F, ar), bond#2(A, F, G, ar)	10	9
dock(A) :- bond#3(A, B, C, 1), atom(A, B, o), bond#1(A, D, B, 1), bond#1(A, E, C, ar), bond#1(A, F, D, 1), bond#1(A, G, E, ar)	10	9
dock(A) :- atom#1(A, B, o), atom#1(A, C, h), bond#1(A, D, B, ar), bond#1(A, E, C, 1), bond#2(A, E, F, 1), ring#2(A, G, F, 5, ar)	10	7
dock(A) :- atom#1(A, B, o), atom#1(A, C, f), bond#1(A, D, B, 1), bond#2(A, C, E, 1), bond#2(A, E, F, 1), bond#2(A, F, G, 1)	11	4
dock(A) :- bond#3(A, B, C, 1), atom(A, C, n), bond#1(A, D, B, 1), bond#1(A, E, C, 1), bond#1(A, F, D, 2), ring#2(A, G, E, 6, not_ar)	11	10
dock(A) :- bond#3(A, B, C, ar), atom(A, B, n), bond#1(A, D, C, ar), bond#1(A, E, D, ar), bond#2(A, C, F, ar), bond#2(A, F, G, ar)	18	10
dock(A) :- atom#1(A, B, n), atom#1(A, C, o), bond#1(A, D, C, ar), bond#1(A, E, B, 1), bond#1(A, F, E, ar)	11	10
dock(A) :- bond#3(A, B, C, 1), atom(A, B, s), bond#1(A, D, C, ar), bond#1(A, E, D, ar), ring#2(A, F, C, 5, ar)	11	8
dock(A) :- bond#3(A, B, C, 1), atom(A, C, h), bond#1(A, D, B, ar), bond#1(A, E, D, ar), bond#1(A, F, D, 1), ring#2(A, G, E, 5, ar)	10	8
dock(A) :- atom#1(A, B, c), atom#1(A, C, h), bond#1(A, D, C, 1), bond#2(A, B, E, 1), ring#2(A, F, D, 5, not_ar), ring#2(A, G, E, 5, ar)	10	9
dock(A) :- bond#3(A, B, C, 1), atom(A, B, c), bond#1(A, D, C, ar), ring#2(A, E, C, 5, ar), ring#2(A, F, D, 6, ar)	10	8
dock(A) :- bond#3(A, B, C, 1), atom(A, C, h), bond#1(A, D, B, 1), bond#2(A, D, E, 1), bond#2(A, E, F, ar)	12	10
dock(A) :- bond#3(A, B, C, 1), atom(A, C, n), bond#1(A, D, B, 1), bond#1(A, E, C, ar), bond#1(A, F, E, ar), bond#2(A, F, G, 1)	10	8
dock(A) :- atom#1(A, B, n), atom#1(A, C, o), bond#1(A, D, C, ar), bond#2(A, B, E, ar), bond#2(A, E, F, ar), ring#2(A, G, F, 5, ar)	12	9

Fig. 4 Rule list