



2017.9.8 @日本応用数理学会 2017年度 年会 [機械学習研究部会 OS]

# グラフデータの機械学習における 特徴表現の設計と学習

瀧川 一学

北海道大学・JSTさきがけ

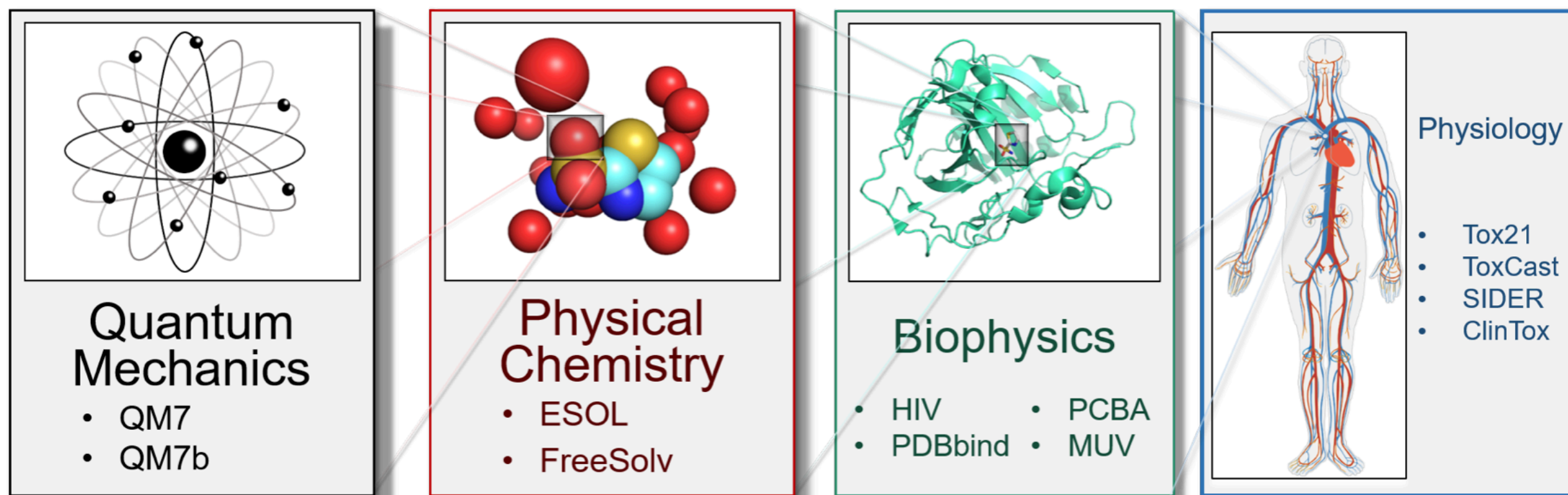
[takigawa@ist.hokudai.ac.jp](mailto:takigawa@ist.hokudai.ac.jp)

- 背景：有機低分子の活性・物性の予測モデリング
- 多数のグラフからの教師つき学習
  - スペース線形学習(Takigawa+ 2017)
  - 勾配ブースティングによる回帰森(Yokoyama+ Submitted)
- 部分構造指示子による0-1ブール変数に対する仮説空間
  - 相互作用と線形 vs 非線形：Graph-XOR問題
- 低分子化合物のグラフエンコードと特徴量計算
  - Molecular Graph, Invariants, Input Featurization
  - Circular FingerprintとMorgan Algorithm
- 最近の話題：量子化学計算近似・材料Informatics
  - Neural Message Passing* (Google Brain/DeepMind/Basel-U)

- |   |             |
|---|-------------|
| <ul style="list-style-type: none"><li>• 背景：有機低分子の活性・物性の予測モデリング</li><li>• 多数のグラフからの教師つき学習<ul style="list-style-type: none"><li>• スペース線形学習(Takigawa+ 2017)</li><li>• 勾配ブースティングによる回帰森(Yokoyama+ Submitted)</li></ul></li><li>• 部分構造指示子による0-1ブール変数に対する仮説空間<ul style="list-style-type: none"><li>• 交互作用と線形 vs 非線形：Graph-XOR問題</li></ul></li></ul>                                    | 1部<br>(10分) |
| <ul style="list-style-type: none"><li>• 低分子化合物のグラフエンコードと特徴量計算<ul style="list-style-type: none"><li>• Molecular Graph, Invariants, Input Featurization</li><li>• Circular FingerprintとMorgan Algorithm</li></ul></li><li>• 最近の話題：量子化学計算近似・材料Informatics<ul style="list-style-type: none"><li>• <i>Neural Message Passing</i> (Google Brain/DeepMind/Basel-U)</li></ul></li></ul> | 2部<br>(5分)  |

医薬品, 細胞制御剤, 有機材料, 環境物質, 殺虫剤, 除草剤, 化粧品, 食品, ...

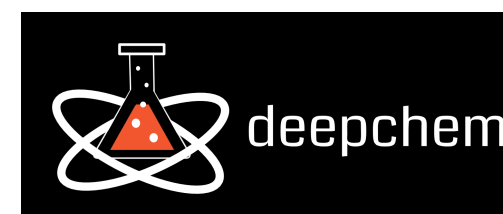
- 化学構造データセットは、数値ではなく「構造(形・関係)」のデータで背後の法則性理解が難しい...😞
- いろんなレベルの活性・物性に対するタスクが研究されている。



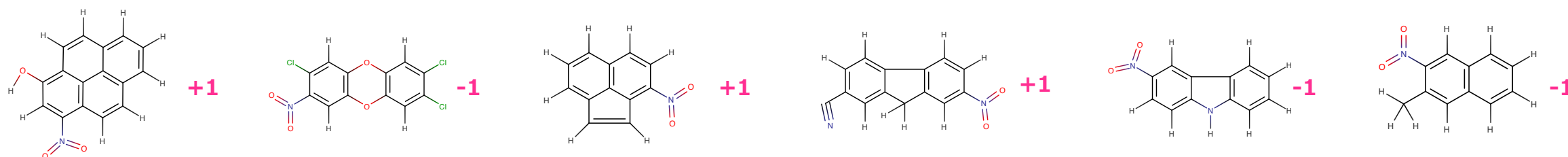
*MoleculeNet: A Benchmark for Molecular Machine Learning*

<https://arxiv.org/abs/1703.00564>

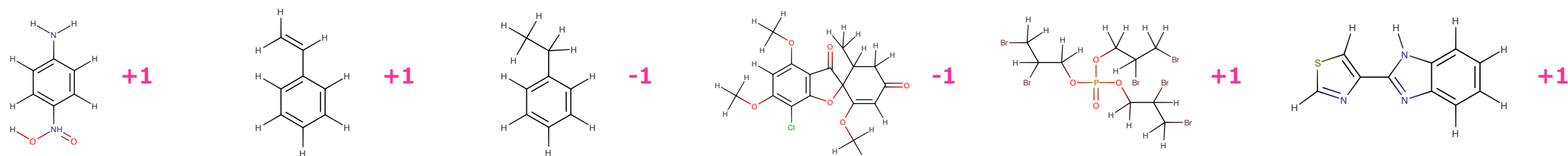
<https://github.com/deepchem/deepchem> (<https://deepchem.io/>)



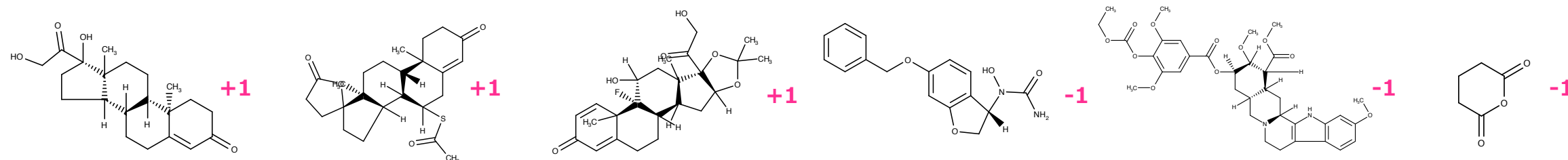
- mutagenic potency (変異原性試験 Ames test)



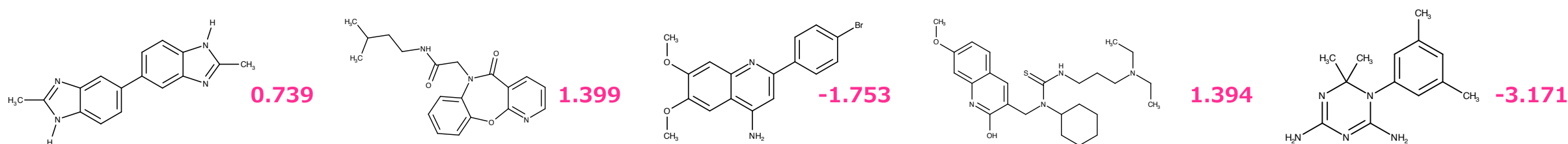
- carcinogenic potency (環境物質の発がん性試験 in vivo)



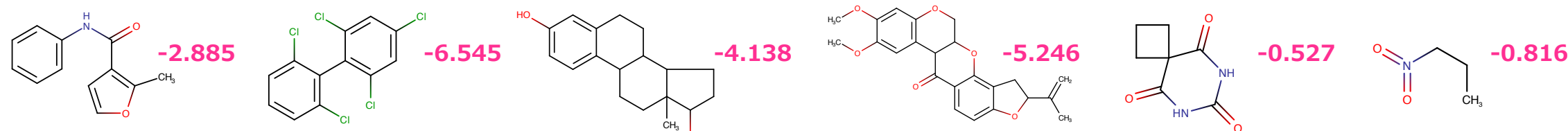
- endocrine disruption (環境物質の内分泌攪乱性試験)



- growth inhibition (抗マラリア原虫性やガン細胞成長阻害性 in vitro)



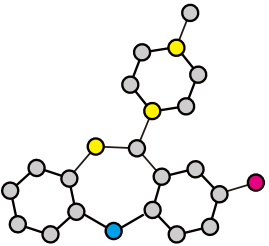
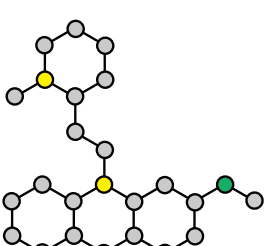
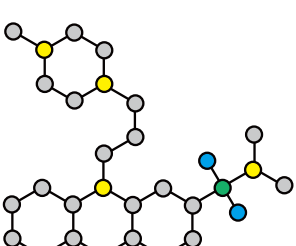
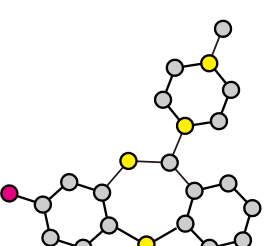
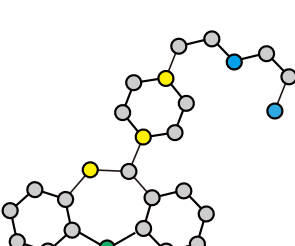
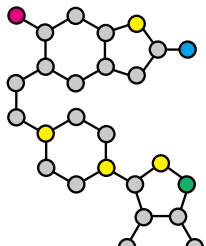
- aqueous solubility (水溶解度)



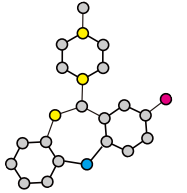
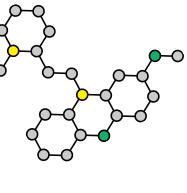
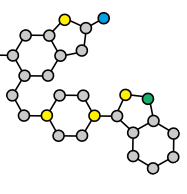
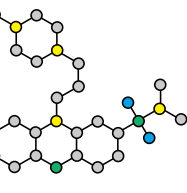
関心：グラフと関連値の対応(関数)の推定 (有限見本対からの教師つき学習)

有限見本対  $\{(G_i, y_i)\}_{i=1}^n$  から  $y \approx \mu(G)$  となるようなモデル  $\mu$  を構成したい

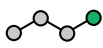

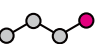



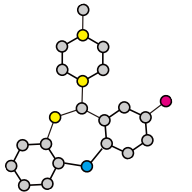
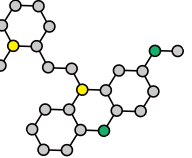

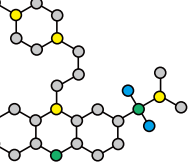


$y_1$	$y_2$	$y_3$	$y_4$	$y_5$		$y_n$
0.1	0.7	1.2	0.2	1.3	...	0.9
$G_1$	$G_2$	$G_3$	$G_4$	$G_5$		$G_n$
					...	

**注意！** 大きなネットワークが一つあり各頂点での関連値を予測する問題とは異なる

$y$	$G$
0.1	
0.7	
0.9	
⋮	⋮
1.2	

部分グラフ特徴

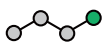

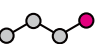



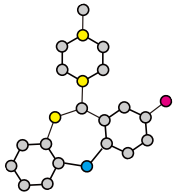
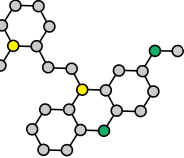
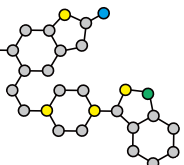
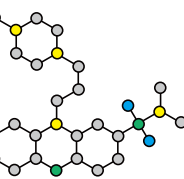
		部分グラフ特徴						
		$g_1$	$g_2$	$g_3$	$g_4$	$g_5$	$g_6$	
$y$	$G$							...
0.1		0	0	1	1	1	0	...
0.7		1	0	0	0	0	1	...
0.9		1	1	0	1	1	0	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1.2		1	0	1	1	1	0	...

部分グラフ同型のindicator

$$\mathbb{I}(G_i \supseteq g_j) = \begin{cases} 0 \\ 1 \end{cases}$$



部分グラフ特徴

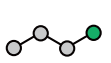

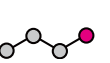


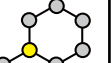
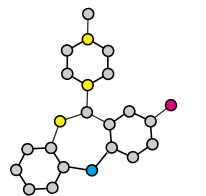
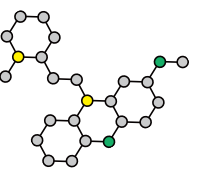
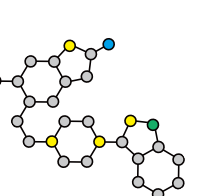
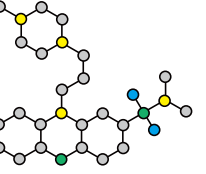
		部分グラフ特徴						
		$g_1$	$g_2$	$g_3$	$g_4$	$g_5$	$g_6$	
$y$	$G$							...
0.1		0	0	1	1	1	0	...
0.7		1	0	0	0	0	1	...
0.9		1	1	0	1	1	0	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1.2		1	0	1	1	1	0	...

部分グラフ同型のindicator

$$\mathbb{I}(G_i \supseteq g_j) = \begin{cases} 0 \\ 1 \end{cases}$$

- 普通は事前に特徴は与えられないので  
必要な特徴を探索しながら同時学習

部分グラフ特徴

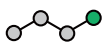

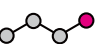



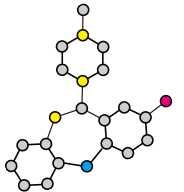
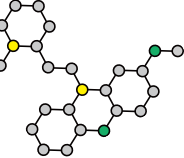
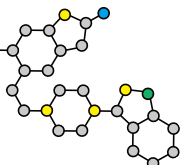
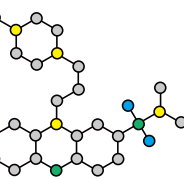
		部分グラフ特徴						
		$g_1$	$g_2$	$g_3$	$g_4$	$g_5$	$g_6$	
$y$	$G$							...
0.1		0	0	1	1	1	0	...
0.7		1	0	0	0	0	1	...
0.9		1	1	0	1	1	0	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱
1.2		1	0	1	1	1	0	...

部分グラフ同型のindicator

$$\mathbb{I}(G_i \supseteq g_j) = \begin{cases} 0 \\ 1 \end{cases}$$

- 普通は事前に特徴は与えられないので  
必要な特徴を探索しながら同時学習
- 各々の入力グラフGは a bag of  
subgraph features で表現される

部分グラフ特徴

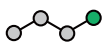

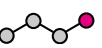


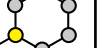
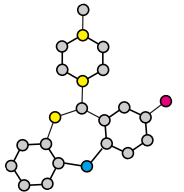
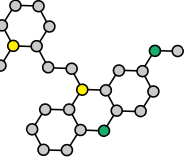
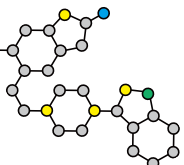
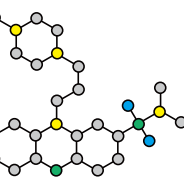
		$g_1$	$g_2$	$g_3$	$g_4$	$g_5$	$g_6$	
$y$	$G$							...
0.1		0	0	1	1	1	0	...
0.7		1	0	0	0	0	1	...
0.9		1	1	0	1	1	0	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱
1.2		1	0	1	1	1	0	...

部分グラフ同型のindicator

$$\mathbb{I}(G_i \supseteq g_j) = \begin{cases} 0 \\ 1 \end{cases}$$

- 普通は事前に特徴は与えられないので  
必要な特徴を探索しながら同時学習
- 各々の入力グラフGは a bag of  
subgraph features で表現される
- indicatorで考えると0-1ブール変数  
(変数候補の数は組合せ爆発的に巨大)

部分グラフ特徴

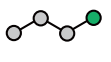

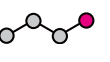



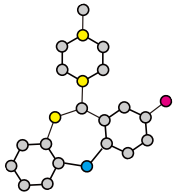
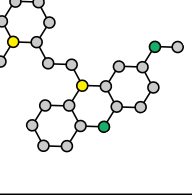
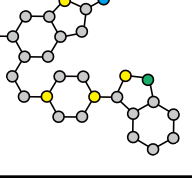
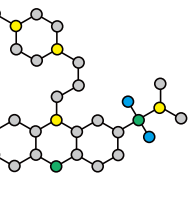
		部分グラフ特徴						
		$g_1$	$g_2$	$g_3$	$g_4$	$g_5$	$g_6$	
$y$	$G$							...
0.1		0	0	1	1	1	0	...
0.7		1	0	0	0	0	1	...
0.9		1	1	0	1	1	0	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1.2		1	0	1	1	1	0	...

部分グラフ同型のindicator

$$\mathbb{I}(G_i \supseteq g_j) = \begin{cases} 0 \\ 1 \end{cases}$$

- 普通は事前に特徴は与えられないので  
必要な特徴を探索しながら同時学習
- 各々の入力グラフGは a bag of  
subgraph features で表現される
- indicatorで考えると0-1ブール変数  
(変数候補の数は組合せ爆発的に巨大)
- 特徴ベクトルが取り得る値は有限  
(超高次ブール超立方体の端点のみ)

部分グラフ特徴

$y$	$G$	部分グラフ特徴						...
		$g_1$	$g_2$	$g_3$	$g_4$	$g_5$	$g_6$	
								...
0.1		0	0	1	1	1	0	...
0.7		1	0	0	0	0	1	...
0.9		1	1	0	1	1	0	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱
1.2		1	0	1	1	1	0	...

部分グラフ同型のindicator

$$\mathbb{I}(G_i \supseteq g_j) = \begin{cases} 0 \\ 1 \end{cases}$$

- 普通は事前に特徴は与えられないので  
必要な特徴を探索しながら同時学習
- 各々の入力グラフGは a bag of  
subgraph features で表現される
- indicatorで考えると0-1ブール変数  
(変数候補の数は組合せ爆発的に巨大)
- 特徴ベクトルが取り得る値は有限  
(超高次ブール超立方体の端点のみ)
- グラフ包含関係による特殊な相関

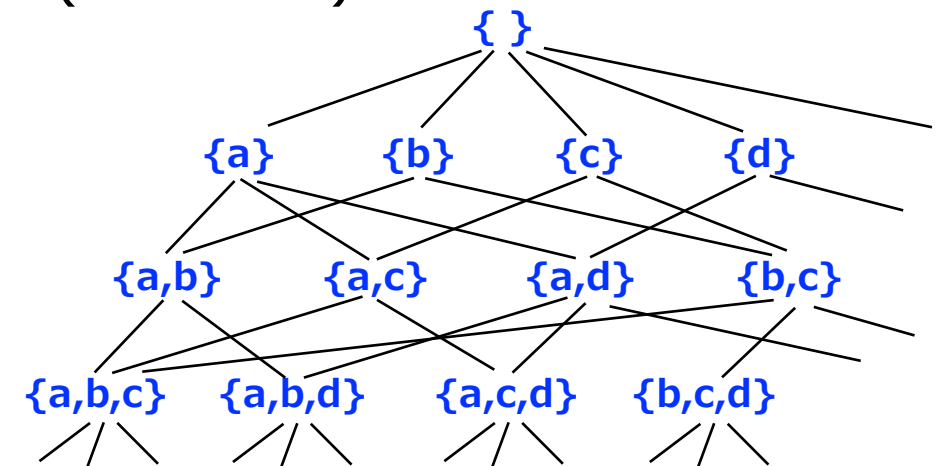
部分アイテムセット特徴

		$x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6$						
$y$	$X$	$\{a\}$	$\{a,d\}$	$\{b,e\}$	$\{b\}$	$\{f,g\}$	$\{h\}$	...
0.1	$\{b,e,f,g\}$	0	0	1	1	1	0	...
0.7	$\{a,g,h\}$	1	0	0	0	0	1	...
0.9	$\{a,b,d,f,g\}$	1	1	0	1	1	0	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$
1.2	$\{a,b,e,f,g\}$	1	0	1	1	1	0	...

※単純には各アイテムのindicator, ただし...(後述)

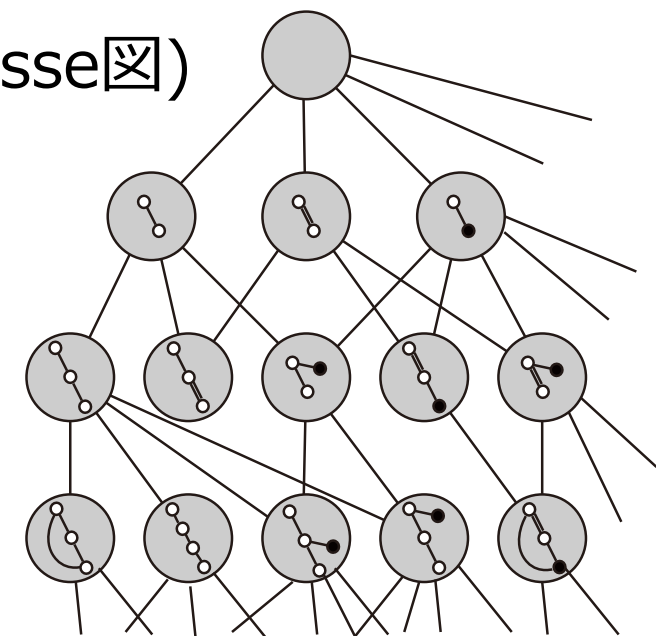
アイテムセットの探索空間

(Hasse図)



部分グラフの探索空間

(Hasse図)



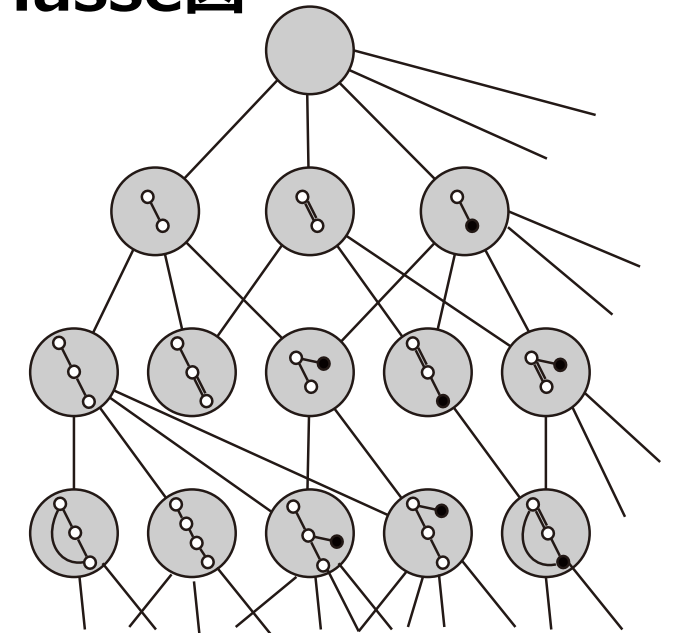
- Kudo+ NIPS 2004

- 各部分グラフ特徴のindicatorとその否定を**弱学習器**とみなす。(実際には定数項処理のため $\{-1, 1\}$ で対照化)
- 最も誤差を減らす部分グラフ特徴を探索する作業が**弱学習**に相当 (否定も1-indicatorで書けるため)
- 上の探索をBranch & Boundで行うための枝刈り条件を導出
- AdaBoost(Freund & Schapire 1997)を用いて**弱学習器の加法的アンサンブル**を構成

- Saigo+ Mach Learn 2009 (gBoost法)

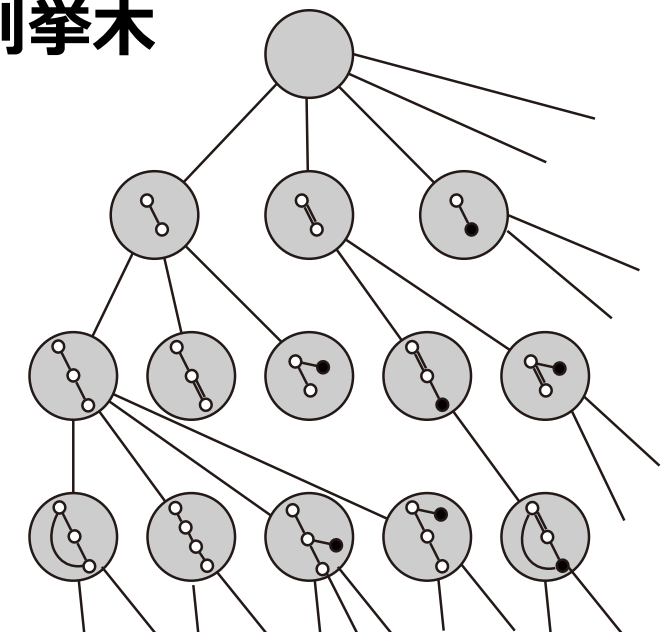
- LPBoost(Demiriz+ 2002)を用いて**弱学習器の加法的アンサンブル**を改良 (totally corrective boosting)
- LPBoost: 学習問題が線形計画で定式化できる場合、線形計画を列生成法で解く操作がboostingとみなせる

Hasse図



深さ優先探索

列挙木



**問題：既存手法は(一見わかりづらいが)近似できる関数が「線形」のみ (線形の仮説空間)**

→ Decision Stumpのブースティングは**実数特徴なら非線形**な仮説を生成できるが  
この問題では全て**0-1ブール特徴**なので加法的に足すと**線形な仮説**に帰着してしまう



**問題：既存手法は(一見わかりづらいが)近似できる関数が「線形」のみ (線形の仮説空間)**

→ Decision Stumpのブースティングは**実数特徴なら非線形**な仮説を生成できるが  
この問題では全て**0-1ブール特徴**なので加法的に足すと**線形な仮説**に帰着してしまう

① 線形モデルでどこまで一般化できるか？ (Takigawa & Mamitsuka TPAMI 2017)

- 指数個ある部分グラフ特徴を妥当なところのみ探索しながら、同時にモデルの学習も行う方式は、**任意の損失関数のL1とL2の正則化つき最小化問題**に適用可
- Block-Coordinate Gradient Descent (Tseng+ 2006)という最適化法を部分グラフを探索しながら行う。L2正則化つけるなら非常にtrickyな枝刈りが必要。

**問題：既存手法は(一見わかりづらいが)近似できる関数が「線形」のみ (線形の仮説空間)**

→ Decision Stumpのブースティングは**実数特徴なら非線形**な仮説を生成できるが  
この問題では全て**0-1ブール特徴**なので加法的に足すと**線形な仮説**に帰着してしまう

① 線形モデルでどこまで一般化できるか？ (Takigawa & Mamitsuka TPAMI 2017)

- 指数個ある部分グラフ特徴を妥当なところのみ探索しながら、同時にモデルの学習も行う方式は、**任意の損失関数のL1とL2の正則化つき最小化問題**に適用可
- Block-Coordinate Gradient Descent (Tseng+ 2006)という最適化法を部分グラフを探索しながら行う。L2正則化つけるなら非常にtrickyな枝刈りが必要。

② 非線形な仮説も生成できるようにするには？ (Yokoyama+ Submitted)

- **0-1変数の交互作用を考えれば良いので(Stumpじゃなくて)Regression Treeの学習を実現(枝刈り条件を導出) + それを勾配ブースティングでアンサンブル学習**
- 何もコントロールしないとオーバーフィットしてしまうため、各Treeのサイズや分割条件、ブースティングのStepsizeなどで適切にComplexityをコントロール

Regression Treeで仮説空間を非線形にすると本当に良いことがあるのか？？

d次元の0-1ブール特徴  $\mathbf{x} = (x_1, x_2, \dots, x_d) \in \{0, 1\}^d$  によるRegression Treeは  
**擬ブール関数**  $f : \{0, 1\}^d \rightarrow \mathbb{R}$ 。擬ブール関数は一意的なmulti-linear polynomialを持つ  
(Hammer+ 1963)

$$f(\mathbf{x}) = a + \sum_i a_i x_i + \sum_{i < j} a_{ij} x_i x_j + \sum_{i < j < k} a_{ijk} \underbrace{x_i x_j x_k}_{\substack{\downarrow \\ i, j, k \text{ の共起のindicator}}} + \dots$$

## Regression Treeで仮説空間を非線形にすると本当に良いことがあるのか？？

d次元の0-1ブール特徴  $\mathbf{x} = (x_1, x_2, \dots, x_d) \in \{0, 1\}^d$  によるRegression Treeは  
**擬ブール関数**  $f : \{0, 1\}^d \rightarrow \mathbb{R}$ 。擬ブール関数は一意的なmulti-linear polynomialを持つ  
(Hammer+ 1963)

$$f(\mathbf{x}) = a + \sum_i a_i x_i + \sum_{i < j} a_{ij} x_i x_j + \sum_{i < j < k} a_{ijk} \underbrace{x_i x_j x_k}_{\substack{\downarrow \\ i, j, k \text{ の共起のindicator}}} + \dots$$

この性質より例えば次のことが分かる。

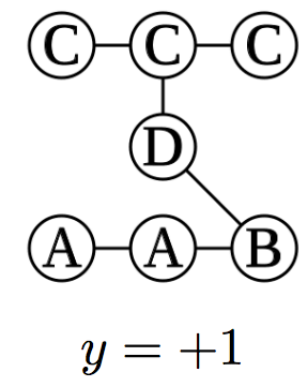
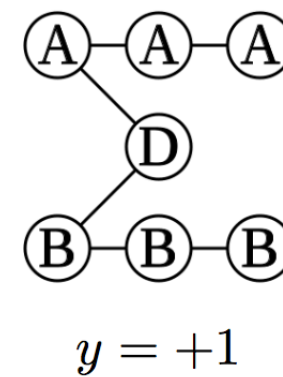
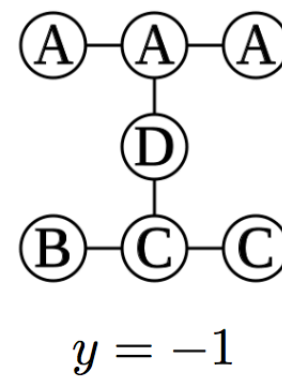
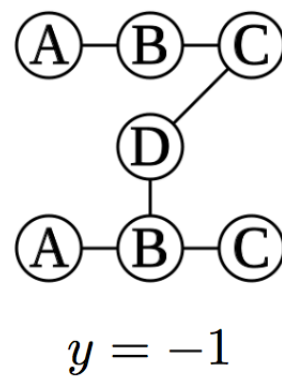
- **部分アイテムセット特徴**を**全て**候補に考える場合、その特徴で**非線形なモデル**を作っても、**線形な仮説空間に帰着**する。(線形モデルで全て表現できる)
- **部分グラフ特徴**は、通常は**連結な部分グラフ**しか探索しないため(効率的に探索ができないため)、**非線形なモデル**がなす仮説空間は一般には**線形仮説空間より大きい**  
→ **グラフの場合で、線形仮説空間では学習に失敗する例を具体的に構成できるか？**

7頂点・6辺の1035個のグラフ (506が正例 +1, 529が負例 -1)

Group 1	Group 2
Ⓐ-Ⓐ-Ⓐ	Ⓑ-Ⓑ-Ⓑ
ⓒ-ⓒ-ⓒ	Ⓐ-Ⓐ-Ⓑ
Ⓐ-Ⓑ-Ⓑ	Ⓐ-Ⓑ-Ⓐ
Ⓑ-Ⓐ-Ⓑ	Ⓑ-Ⓑ-ⓒ
Ⓑ-ⓒ-ⓒ	Ⓑ-ⓒ-Ⓑ
ⓒ-Ⓑ-ⓒ	Ⓐ-Ⓐ-ⓒ
Ⓐ-ⓒ-ⓒ	Ⓐ-ⓒ-Ⓐ
ⓒ-Ⓐ-ⓒ	Ⓐ-Ⓑ-ⓒ
Ⓐ-ⓒ-Ⓑ	Ⓑ-Ⓐ-ⓒ

2つパーツを選んで頂点Dでランダムに結線してグラフを作成

- パーツが両方同じグループから来ていたら -1
- パーツが各々違うグループから来ていたら +1

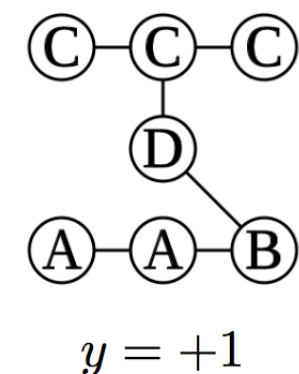
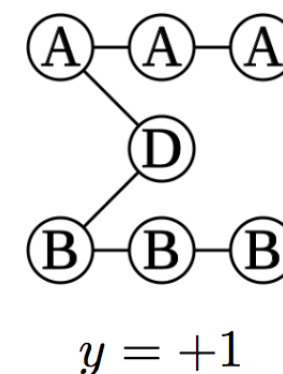
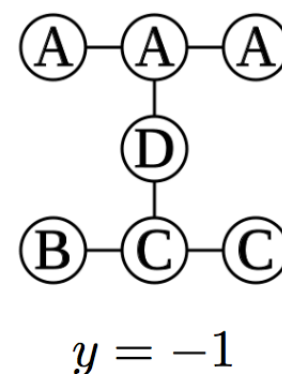
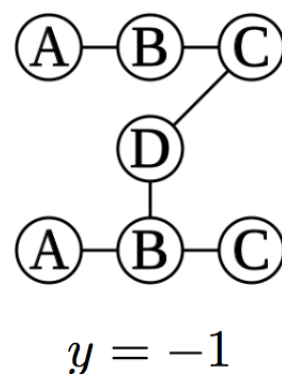


7頂点・6辺の1035個のグラフ (506が正例 +1, 529が負例 -1)

Group 1	Group 2
Ⓐ-Ⓐ-Ⓐ	Ⓑ-Ⓑ-Ⓑ
ⓒ-ⓒ-ⓒ	Ⓐ-Ⓐ-Ⓑ
Ⓐ-Ⓑ-Ⓑ	Ⓐ-Ⓑ-Ⓐ
Ⓑ-Ⓐ-Ⓑ	Ⓑ-Ⓑ-ⓒ
Ⓑ-ⓒ-ⓒ	Ⓑ-ⓒ-Ⓑ
ⓒ-Ⓑ-ⓒ	Ⓐ-Ⓐ-ⓒ
Ⓐ-ⓒ-ⓒ	Ⓐ-ⓒ-Ⓐ
ⓒ-Ⓐ-ⓒ	Ⓐ-Ⓑ-ⓒ
Ⓐ-ⓒ-Ⓑ	Ⓑ-Ⓐ-ⓒ

2つパーツを選んで頂点Dでランダムに結線してグラフを作成

- パーツが両方同じグループから来ていたら -1
- パーツが各々違うグループから来ていたら +1



**結果：**線形(提案法でstump, 既存法gBoost, 列挙特徴でLogistic回帰)だとコケるが  
非線形な仮説空間から学習すれば上手く解ける (下記 2-fold CVでの精度)

Nonlinear models		Linear models		
Proposed	RF	Proposed ( $d1$ )	gBoost	LR
<b>100.0*</b>	<b>94.3</b>	64.3	70.0	53.9
( $m0\ x2\ d2\ \eta0.7\ k221$ )	( $m0\ x2\ f40\ k300\ d\infty$ )	( $m0\ x6\ d1\ \eta0.7\ k26$ )	( $m0\ x6\ \nu0.01$ )	( $m0\ x2$ )

The best result is highlighted in **bold\***, the second in **bold**.



活性予測の実データでも非線形の方が良いものの思ったより線形で十分？

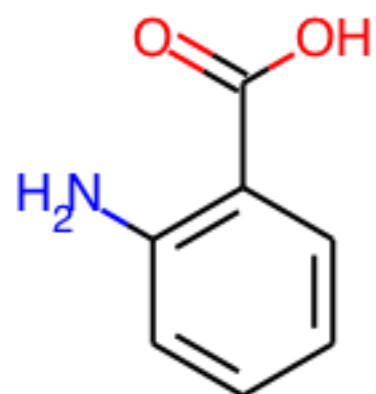
	CPDB ACC	AUC	Mutag ACC	AUC	NCI1 ACC	AUC	NCI47 ACC	AUC
<b>(Nonlinear models)</b>								
Proposed	<b>79.3*</b> (±4.8)	<b>84.5</b> (±3.6)	87.8 (±6.6)	91.6 (±6.3)	<b>84.7</b> (±1.7)	<b>90.8</b> (±1.3)	<b>84.5</b> (±1.7)	<b>90.3</b> (±1.1)
RF	77.6 (±5.6)	83.3 (±4.5)	<b>88.2</b> (±8.2)	<b>93.9</b> (±4.7)	<b>85.8*</b> (±2.6)	<b>92.1*</b> (±1.4)	<b>85.9*</b> (±2.0)	<b>91.7*</b> (±1.5)
<b>(Linear models)</b>								
gBoost	77.1 (±2.7)	73.6 (±4.9)	<b>91.4*</b> (±5.8)	<b>93.9</b> (±5.0)	82.7 (±2.2)	83.9 (±2.2)	81.3 (±1.4)	81.8 (±2.6)
LR	76.5 (±4.2)	84.2 (±3.7)	87.1 (±7.3)	<b>95.1*</b> (±4.4)	83.7 (±1.3)	<b>90.8</b> (±1.3)	84.2 (±1.5)	90.0 (±1.2)
MGK [1]	76.5	75.6	80.8	90.1	-	-	-	-
freqSVM [1]	77.8	<b>84.5</b>	80.8	90.6	-	-	-	-
gBoost [1]	<b>78.8</b>	<b>85.4*</b>	85.2	92.6	-	-	-	-
WL shortest path [2]	-	-	83.7	-	84.5	-	-	-
Random walk [2]	-	-	80.7	-	64.3	-	-	-
Shortest path [2]	-	-	87.2	-	73.4	-	-	-

The best result is highlighted in **bold\***, the second in **bold**.

[1] Saigo et al. (2009), [2] Shervashidze et al. (2011)

- グラフデータの性質と有効な学習戦略の関係をもう少し理解したい…
- 活性・物性予測の場合、グラフのエンコードの仕方もかなり重要…

例) アントラニル酸

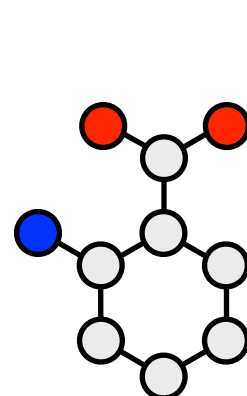


Structural Formula

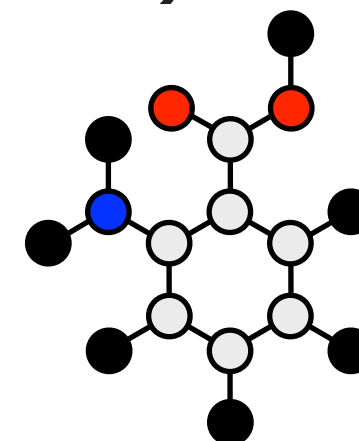
分子グラフへ  
エンコーディング



分子グラフ (トポロジ)



Implicit Hydrogens

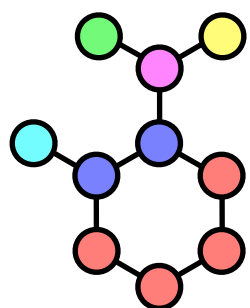


Explicit Hydrogens

頂点(atom)のラベル付けの例 (RDKit defaults) ※辺(atom-pair)もラベル付可

ECFP-like (Atom Invariants)

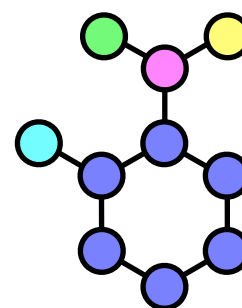
- By Connectivity: atomic number, total degree, #Hs, formal charge, isotope, inRing



	Yellow	Green	Cyan	Blue	Magenta	Red
atomic number	8	8	7	6	6	6
total degree	2	1	3	3	3	3
#Hs	1	0	2	0	0	1
formal charge	0	0	0	0	0	0
isotope	0	0	0	0	0	0
in Ring?	0	0	0	1	0	1

FCFP-like (Feature Invariants)

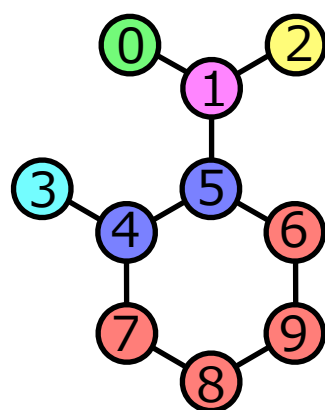
- By Chemical Features: Donor, Acceptor, Aromatic, Halogen, Basic, Acidic



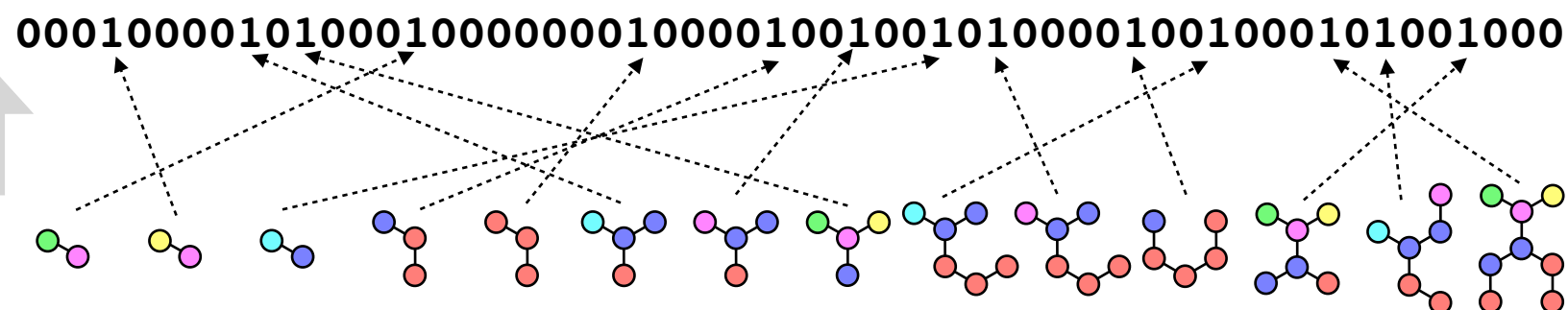
	Yellow	Green	Cyan	Blue	Magenta
Donor	1	0	1	0	0
Acceptor	0	1	1	0	0
Aromatic	0	0	0	1	0
Halogen	0	0	0	0	0
Basic	0	0	1	0	0
Acidic	0	0	0	0	1



分子グラフ

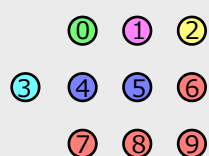


固定長の0-1特徴ベクトル (fingerprint)

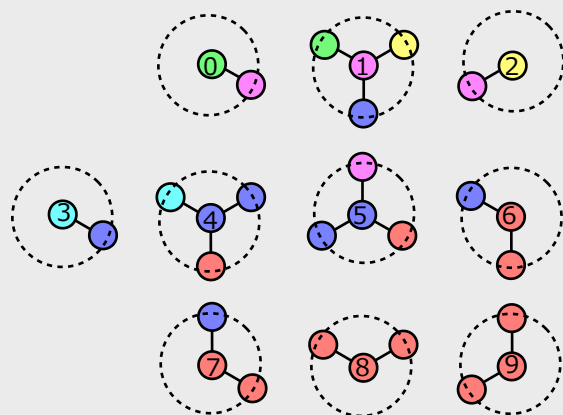


各subgraphの整数IDをmodして得た桁のbitを1に

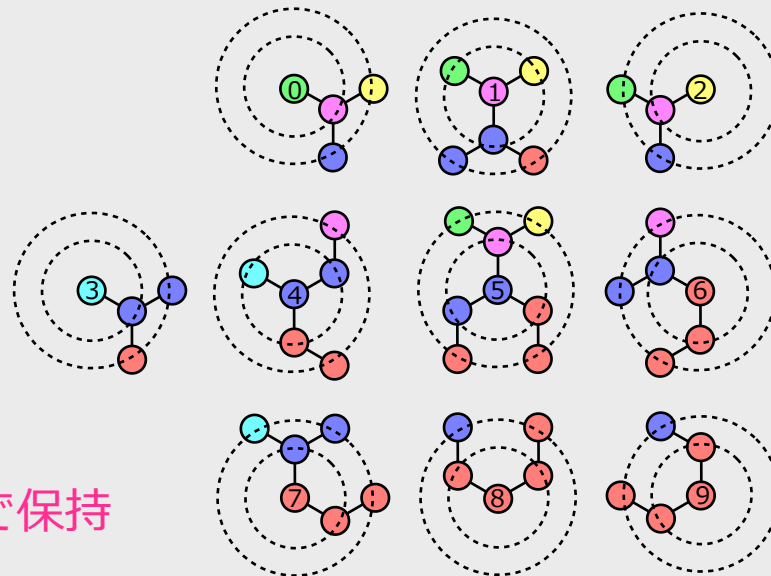
Layer-0  
(diameter 0)



Layer-1  
(diameter 2)



Layer-2  
(diameter 4)

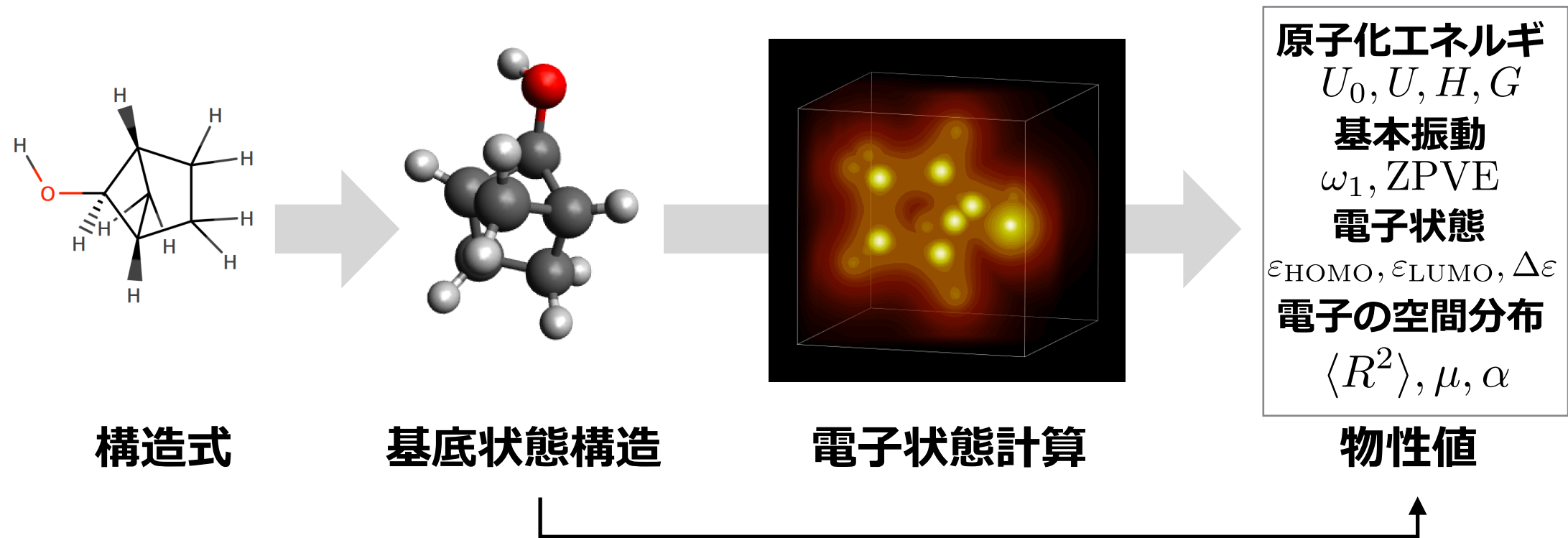


実際には各subgraphは整数IDで保持

※ECFP4(diameter 4まで), ECFP6(diameter 6まで)がよく使われる

- Layer-0では、各頂点のnode invariantsの配列をhashして得た整数IDをその頂点に付与
  - node invariantsにはその頂点の周辺情報が暗に含まれる
- [Morgan Algorithmの変種]
- Layer-1では、各頂点について、隣り合う頂点の整数IDの配列をhashして得た新たな整数IDで更新
  - Layer-2以降ではLayer-1で更新された各々の頂点の整数IDを使って同じように近傍で整数IDを更新

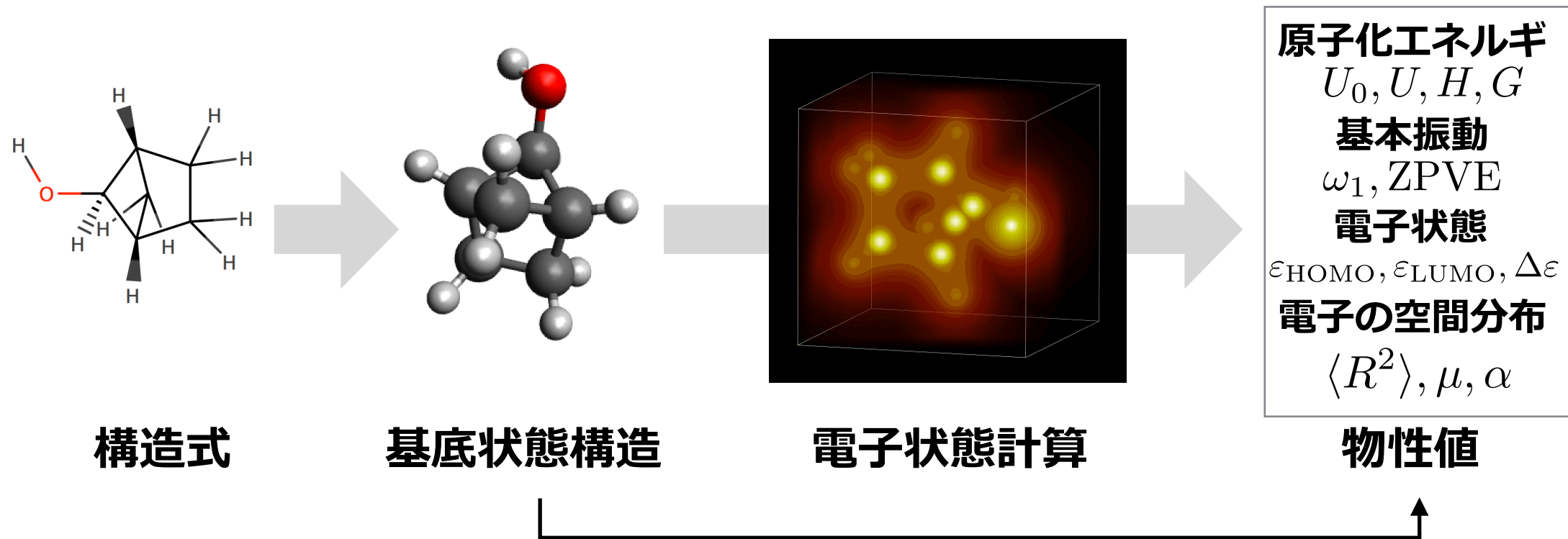
## 量子化学計算 (第一原理計算, つまりSchrödinger方程式の数値的求解)



Question: この計算を真面目にやると時間が超かかるので  
これをデータ駆動でquickに高精度近似できないか?

- 生物活性などは分子単独の性質ではなく、基本的に何らかの相手(相互作用)があることが前提だが、この設定では物質そのもので決まる値が対象
- 各原子の3D座標を使える (逆に言えば3D座標が変わると変化する値が対象)
- 理論計算値の予測のデータセットの場合、ノイズを含まないかなりクリーンなデータ
- 生物活性と異なり、時間さえかけられるのなら対象値を計算する「原理」が存在する

## 量子化学計算 (第一原理計算, つまりSchrödinger方程式の数値的求解)



Question: この計算を真面目にやると時間が超かかるので  
これをデータ駆動でquickに高精度近似できないか?

- 生物活性などは分子単独の性質ではなく、基本的に何らかの相手(相互作用)があることが前提だが、この設定では物質そのもので決まる値が対象
- 各原子の3D座標を使える (逆に言えば3D座標が変わると変化する値が対象)
- 理論計算値の予測のデータセットの場合、ノイズを含まないかなりクリーンなデータ
- 生物活性と異なり、時間さえかけられるのなら対象値を計算する「原理」が存在する

Googleは2017年4月に2報の論文を発表

① Gilmer+ [Neural Message Passing for Quantum Chemistry](#). arXiv:1704.01212

Neural graph fingerprint (Duvenaud+ 2015), Molecular graph convolutions (Kearns+ 2016), Gated graph sequence NN (Li+ 2016) など任意サイズのグラフを入力にできるニューラルネット学習を抽象化

- Morgan Algorithmの各hashingを学習可能なパラメタを持ったニューラルネット  
で置換する様々なやり方を抽象化した感じ (Network-in-Network的な?)
- 構成要素に何を使うのが良いか色々な組合せを調べ検討した結果、新規なvariantの  
精度が最も良かったという報告
- 入力分子グラフのエッジ特徴に実数特徴を入れている(もはや離散問題ではない!)

② Faber+ [Machine learning prediction errors better than DFT accuracy](#). arXiv:1702.05532

①を使ってQM9(133,885分子)の13物性を従来より300,000倍高速に  
予測計算 (精度は密度汎関数法による電子状態計算と遜色ないか良い)

各頂点は隠れ状態(学習パラメタ)を持つ  $\{h_v^t \mid v \in V(G), t = 1, 2, \dots, T\}$

For each layer  $t = 1, 2, \dots, T$ : (Message Passing Phase)

For each vertex  $v \in V(G)$ :

$$\mathbf{m}_v^{t+1} \leftarrow \sum_{w \in N(v)} \boxed{M_t(\mathbf{h}_v^t, \mathbf{h}_w^t, \mathbf{e}_{vw})}$$

頂点wからvへのメッセージを  
vの近傍N(v)から集約

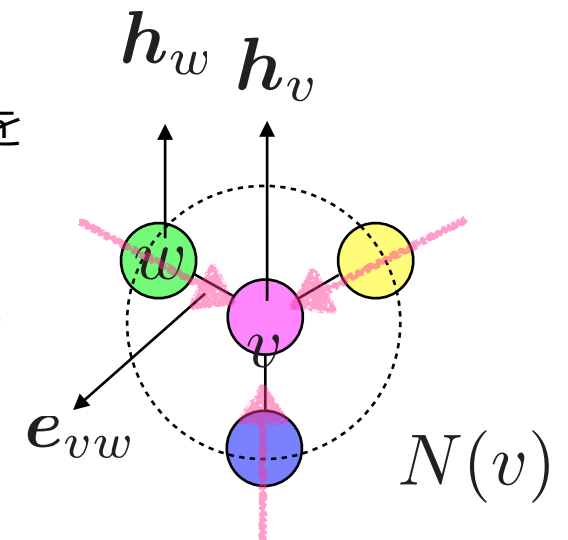
$$\mathbf{h}_v^{t+1} \leftarrow \boxed{U_t(\mathbf{h}_v^t, \mathbf{m}_v^{t+1})}$$

集約したメッセージを用いて  
vの隠れ状態を更新

Compute a feature vector for  $G$  (Readout Phase)

$$\mathbf{y} \leftarrow \boxed{R(\{\mathbf{h}_v^T \mid v \in V(G)\})}$$

各頂点の最終的な隠れ状態を  
統合して出力ベクトルを作る



各頂点は隠れ状態(学習パラメタ)を持つ  $\{h_v^t \mid v \in V(G), t = 1, 2, \dots, T\}$

For each layer  $t = 1, 2, \dots, T$ : (Message Passing Phase)

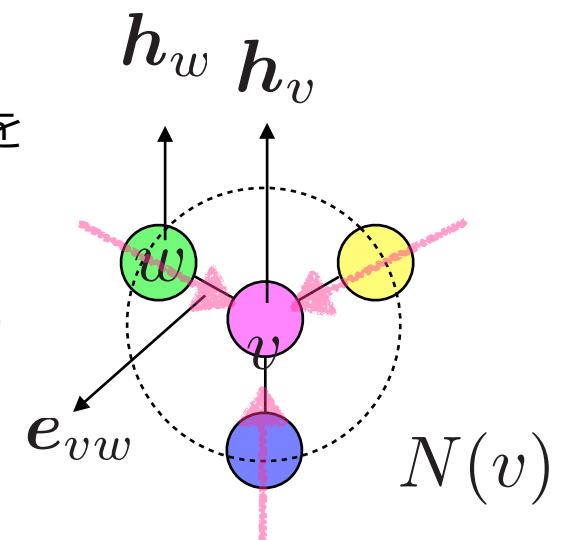
For each vertex  $v \in V(G)$ :

$$m_v^{t+1} \leftarrow \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw})$$

頂点wからvへのメッセージを  
vの近傍N(v)から集約

$$h_v^{t+1} \leftarrow U_t(h_v^t, m_v^{t+1})$$

集約したメッセージを用いて  
vの隠れ状態を更新



Compute a feature vector for  $G$  (Readout Phase)

$$y \leftarrow R(\{h_v^T \mid v \in V(G)\})$$

各頂点の最終的な隠れ状態を  
統合して出力ベクトルを作る

- M, U, Rを微分可能にしておけば、(Morgan algorithmのhash値による各整数ID値に相当する)多次元の隠れ状態hをbackpropで学習できる
- Mにエッジ特徴 $\rightarrow$ hの変換行列を学習するニューラルネットと学習行列による線形変換、UにGated Recurrent Unit (Cho+, 2014)、Rにset2set (Vinyals+, 2015)、という組合せが実験的精度はBestだった (set2seqな気もするが論文にはset2setとある…??)
- Message Passingで近傍からのメッセージの和を取る点とReadoutが肝(この工夫で頂点数が同一でなくてもOKになる)だがこれは近傍情報を潰す(平均化する)のでこれで良いのかはまだ検討の余地あり(?)



- Neural Message Passing, MPNNの論文はICML2017で発表
  - MPNNの枠組のInstanceの検討においてGated Graph Sequence Neural Networks, GGS-NN (Li+ ICLR2016)を非常に重視しているように見える
  - GG-NNの発展手法として、Gated Graph Transformer Neural Network GGT-NN (Johnson 2017)がICLR2017で発表
  - QM9の予測に関しては、Continuous-filter convolutional layersというカスタム層に基づくSchNet (Schütt+ 2017)が提案されており上回る精度の報告もある。K.-R. Müllerグループ+A. Tkatchenkoのjoint work.
- こうした方法では大規模データがあれば物性予測の精度を向上できることが示唆されて来ているが、一方で、こうした緻密化が生物活性予測にも有効かは未だにOpen Question。(ECFPよりもFCFPが良い場合もあるとか)
  - 現時点では、分子グラフに載せる特徴のengineeringが肝になっており、画像の畳み込みのケースほどは、end-to-endとは言えない。

- 背景：有機低分子の活性・物性の予測モデリング
- 多数のグラフからの教師つき学習
  - スペース線形学習(Takigawa+ 2017)
  - 勾配ブースティングによる回帰森(Yokoyama+ Submitted)
- 部分構造指示子による0-1ブール変数に対する仮説空間
  - 相互作用と線形 vs 非線形：Graph-XOR問題
- 低分子化合物のグラフエンコードと特徴量計算
  - Molecular Graph, Invariants, Input Featurization
  - Circular FingerprintとMorgan Algorithm
- 最近の話題：量子化学計算近似・材料Informatics
  - Neural Message Passing* (Google Brain/DeepMind/Basel-U)