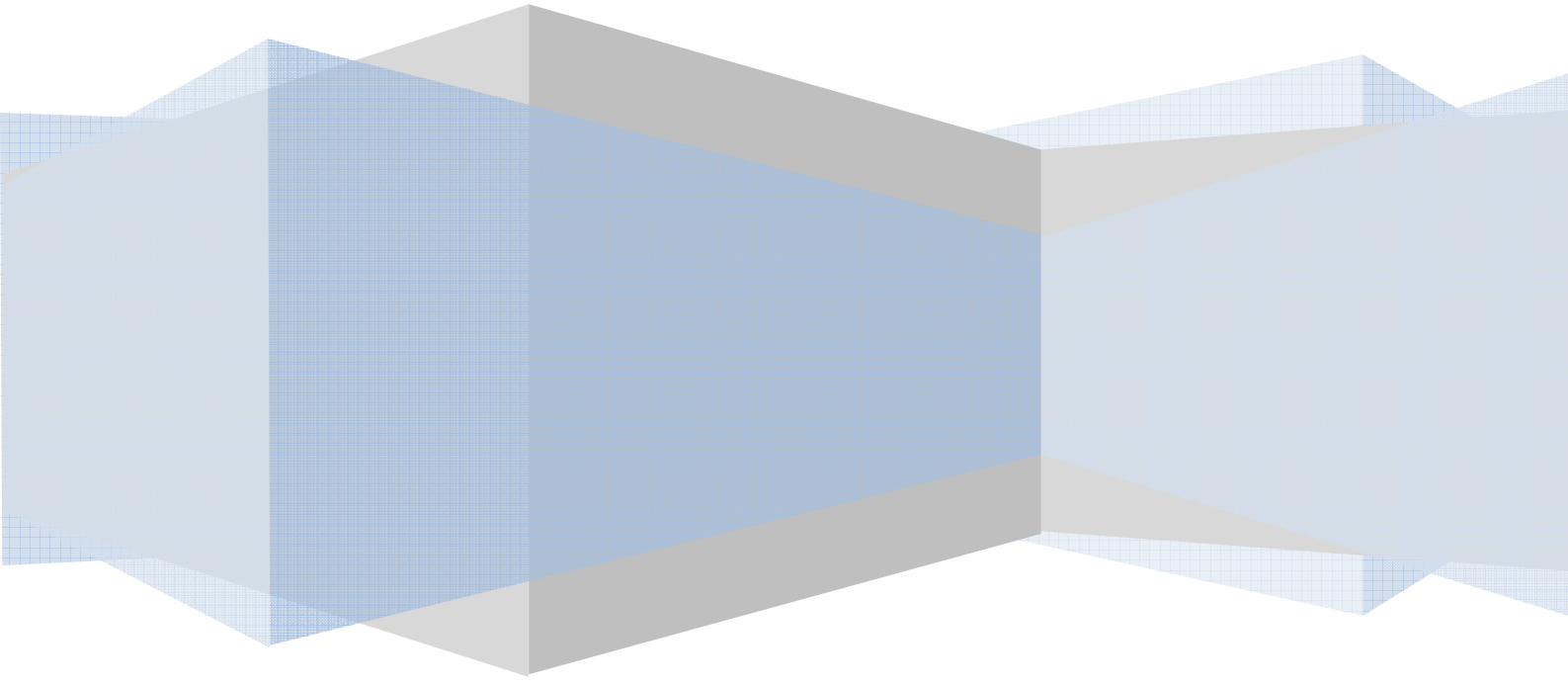
株式会社京都コンステラ・テクノロジーズ

CzeekS マニュアル

2013/2/26



目次

【商標について】

本マニュアルに記載の社名、商品名等は各社の商標または登録商標である場合があります。また、本マニュアルに記載のシステム名、製品名等には、商標表示を付記しておりません。

1. はじめに

近年、ある１つの化合物がたった１つのタンパク質とのみ活性を有するという事は稀なことであるという考え方が常識になって来ています。このような化合物-タンパク質の複雑な関係の情報を我々はケミカルゲノミクス情報と呼称しており、このような情報は ChEMBL 等に代表される化合物の生物活性のデータベースとして整備されつつあります。これらの情報を機械学習によるパターン認識によって未知化合物の活性を予測・スクリーニングする手法が CGBVS（Chemical Genomics-Based Virtual Screening）です。CzeekS はこの CGBVS を行うためのツール群で、以下の機能を提供しています。

・化合物のスコア計算
・CGBVS 学習モデルの作成
・学習モデルの管理機能
・化合物のフィンガープリント（MACCS）の計算
・ターゲット化合物との類似度計算

本マニュアルの 2 章では CzeekS のインストール方法について説明します。3 章ではサンプルデータを使って化合物のスクリーニング方法について説明します。また、発展的な使用方法として化合物の選択性の予測や、ターゲット予測についても説明します。4 章ではサンプルデータを使って学習モデルの構築方法について説明します。5 章はコマンドのレファレンスです。

以下のような計算機環境において CzeekS の利用を想定しています。CzeekS は OpenMP による並列計算に対応していますので、CPU コア数が多い方が効率よく計算することができます。また、計算機は 1 台から CzeekS を利用することができます。

CPU	4 コア以上のマルチコア CPU (Intel, AMD)
メモリ	8GB 以上
HDD	10GB 以上の空き容量
OS	CentOS5.x or 6.x 64bit (Linux カーネル 2.6)
外部ツール	DRAGON6
外部ライブラリ	OpenBabel 2.3.1

サンプルデータの機械学習に掛かる時間（1 ノード）

CPU	スレッド数	メモリ	計算時間
Intel Xeon E5620 × 2	16	24GB	20h 10m
Intel Core i3 550	4	4GB	66h 52m
AMD PhenomⅡ X6 1055T	6	8GB	70h 40m

2. CzeekS のインストールと設定

2-1. アーカイブの展開とライセンスファイルの配置

アーカイブファイル"CzeekS_******.tgz"を以下の様に tar コマンドで展開して下さい。いずれのディレクトリで展開しても構いませんが、/usr/local の下や、czeeks 等のユーザーを作成し、/home/czeeks の下に展開することをお勧めします。尚、本マニュアルでは/home/czeeks 下にファイル展開したと仮定して説明を進めます。

```
$ tar xvfz CzeekS_******.tgz⏎
CGBVS/
CGBVS/exec/
CGBVS/exec/license.dat
CGBVS/exec/cgbvs
CGBVS/exec/calc_dragon.sh
CGBVS/exec/2D_990.drt
CGBVS/exec/calc_FP_MACCS
CGBVS/exec/SVMlearn
CGBVS/exec/protein.lst
・・・
```

展開されるファイルは以下の通りです。添付のライセンスファイル"license.dat"を /home/czeeks/CGBVS/exec 下に上書きコピーして下さい。

```
CGBVS
|-- example                 サンプルデータ等を収めたディレクトリ
|   |-- gpcr.csv            GPCR の記述子ベクトル
|   |-- positive.csv        正例のファイル
|   |-- sample_mols.csv     テスト用化合物の記述子
|   |-- sample_mols.fp      テスト用化合物のフィンガープリント
|   |-- sample_mols.sdf     テスト用化合物の SD ファイル
|   |-- sample_mols.smi     テスト用化合物の SMILES
|   |-- training_mols.csv   学習用サンプル化合物の記述子
|   |-- training_mols.fp    学習用サンプル化合物のフィンガープリント
|   |-- training_mols.sdf   学習用サンプル化合物の SD ファイル
|   `-- training_mols.smi   学習用サンプル化合物の SMILES
`-- exec                    実行ファイル等を収めたディレクトリ
    |-- 2D_990.drt          DRAGON6 用のスクリプトファイル
    |-- SVMlearn            SVM 学習コマンド
    |-- calc_FP_MACCS       MACCS フィンガープリント計算コマンド
    |-- calc_dragon.sh      DRAGON6 実行用スクリプト
    |-- cgbvs               CGBVS 実行コマンド
    |-- license.dat         ライセンスファイル（初期は無効なライセンスです）
    `-- protein.lst         タンパクリスト
```

2-2. 環境変数の設定

アーカイブファイルを展開し、ライセンスファイルを配置した後に、以下のように環境変数を設定して、.bashrc ファイルにも以下の内容を記述して下さい。

```
$ export CGBVS=/home/czeeks/CGBVS/exec⏎
$ export PATH=$PATH:$CGBVS⏎
$ export LD_LIBRARY_PATH=/usr/local/lib:$LD_LIBRARY_PATH⏎
$ export DRAGON6=/usr/local/bin⏎ DRAGON6 がインストールされたパス
```

環境変数 DRAGON6 については、DRAGON6 の実行コマンドである dragon6shell の存在するディレクトリを指定してください。また、ライセンスファイル license.dat を${CGBVS}下以外の場所に配置したい場合は、環境変数 CGBVS_LICENSE にフルパス付きのファイル名を指定してください。

2-3. OpenBabel の設定

CzeekS では化合物のフィンガープリント（MACCS）の計算（calc_FP_MACCS）や SD ファイルから SMILES の生成に OpenBabel を利用しています。インストールされていない場合は以下の手順でインストールしてください。

① cmake のインストール

OpenBabel のコンパイルには cmake が必要です。cmake がインストールされていない場合はインストールしてください。例えば、スーパーユーザーになり"yum install cmake"でインストールすることができます。

② OpenBabel のコンパイルとインストール

OpenBabel はフリーソフト（GPL v2）です。次の URL からダウンロードして下さい。

http://openbabel.org/wiki/Get_Open_Babel

本マニュアルで使用するバージョンは 2.3.1 です。ダウンロードしましたら次にコンパイルします。アーカイブファイルを tar コマンドで展開すると、openbabel-2.3.1 というディレクトリが作成されファイルが展開されます。現在のディレクトリを openbabel-2.3.1 に移して、以下の手順でコンパイルしてください。

```
$ mkdir build⏎          適当なディレクトリを作成
$ cd build⏎
$ cmake ../⏎            cmake コマンドを実行
$ make⏎                 OpenBabel をコンパイル
$ su⏎                   スーパーユーザーになります
# make install⏎         デフォルトのパスにインストール
```

上記の手順は CzeekS を使用する上で、OpenBabel の必要最低限のインストールとなります。コンパイル時の詳細設定などは OpenBabel のマニュアル等を参照してください。

3. 化合物スクリーニングとターゲット予測

3-1. CGBVS モデル

CzeekS にはサンプルのモデルファイルが添付されております。あくまでもサンプルなので、実際のインシリコスクリーニングに用いること避けて下さい。CzeekS ではモデルファイルのファイル名の拡張子は.db となり、以下では“DB ファイル”と呼ぶこともあります。このサンプルモデルは公的データベースの ChMBLE に由来する GPCR のデータを使って構築しており、またモデル構築に用いたデータも添付しています。これらのデータについては 4 章で解説します。

CGBVS ではパターン認識の手法としてサポートベクターマシン（SVM）を用いています。SVM は正例と負例の２クラスを分類する方法で、機械学習するには正負両方のデータが必要です。しかしながら化合物-タンパク質ペアで活性有りという情報は潤沢に存在しても、化合物-タンパク質ペアが実験的に活性無しと確認されている情報は公的データベース等には僅かしか在りません。したがって、負例としての化合物-タンパク質ペアの相互作用情報を仮想的に生成して、機械学習を行っております。仮想的負例は正例ペアをランダムに組み換えることにより生成されております。ランダム性が入るため複数の負例セットで学習モデルを生成して化合物の予測スコアを負例セットの数だけ算出し、最終的にそれらの平均値をスコアとしております。

CzeekS で算出される CGBVS のスコアは 2 種類あります。1 つは SVM の決定関数値の平均値で、-∞~+∞までの範囲をとります。もう一方はこの決定関数値をシグモイド関数で正規化したものの平均値で、0~1 までの範囲をとります。CzeekS では通常は後者の正規化した方のスコアを表示します。このスコアは、ターゲットのタンパク質に対して化合物が活性を持つ“確率値”としての意味を持ちます。つまり、この値と活性値とは比例関係にはありませんのでご注意ください。

上記で説明した CGBVS モデルの情報を”cgbvs status”コマンドで確認することができます。先ず以下のコマンドでサンプルモデルの DB ファイルを確認して下さい。DB ファイルに登録されている化合物数やタンパク質数、学習したモデルについての情報が一覧表示されます。

```
$ cgbvs status gpcr_sample.db⏎
[compund]
# of data = 13838              登録されている化合物数
# of descriptors = 990         化合物記述子の次元数

[protein]
# of data = 859                登録されているタンパク数
# of descriptors = 1497        タンパク記述子の次元数

[interactions]
# of positive interactions = 21761   正例の相互作用情報数
# of negative interactions = 0       負例の相互作用情報数

[details of models]
# of sampled positive interactions = 21761  機械学習に使用した相互作用数
|  id  |   nSV   |  dim  |    C    |  gamma  | accuracy |
|------+---------+-------+---------+---------+----------|
|    1 |   41320 |   462 | 10.0000 |  0.0100 |  81.3377 |
|    2 |   41337 |   462 | 10.0000 |  0.0100 |  81.2850 |
|    3 |   41329 |   462 | 10.0000 |  0.0100 |  81.4914 |
|    4 |   41326 |   462 | 10.0000 |  0.0100 |  81.3329 |
|    5 |   41365 |   462 | 10.0000 |  0.0100 |  81.3393 |
```

最後の学習モデルの詳細についての表について、id はモデルの id 番号で、この例では５つあります。nSV はサポートベクターの数で、C,gamma は SVM のパラメータです。accuracy は各モデルについて交差検定した時の判別の精度を示したものです。

上述の"cgbvs status"コマンドに-p オプションを付けると計算できるタンパク質の一覧表が出力されます。

```
$ cgbvs status -p gpcr_sample.db⏎
protein ID      # of compounds  accession      name
5HT1A_HUMAN           407      P08908  5-hydroxytryptamine receptor 1A
5HT1B_HUMAN           207      P28222  5-hydroxytryptamine receptor 1B
5HT1D_HUMAN           203      P28221  5-hydroxytryptamine receptor 1D
5HT1E_HUMAN            74      P28566  5-hydroxytryptamine receptor 1E
5HT1F_HUMAN           103      P30939  5-hydroxytryptamine receptor 1F
5HT2A_HUMAN           388      P28223  5-hydroxytryptamine receptor 2A
5HT2B_HUMAN           287      P41595  5-hydroxytryptamine receptor 2B
5HT2C_HUMAN           422      P28335  5-hydroxytryptamine receptor 2C
5HT4R_HUMAN           109      Q13639  5-hydroxytryptamine receptor 4
5HT5A_HUMAN           112      P47898  5-hydroxytryptamine receptor 5A
5HT6R_HUMAN           252      P50406  5-hydroxytryptamine receptor 6
5HT7R_HUMAN           227      P34969  5-hydroxytryptamine receptor 7
A4_HUMAN              100      P05067  Amyloid beta A4 protein
AA1R_HUMAN            117      P30542  Adenosine receptor A1
AA2AR_HUMAN           123      P29274  Adenosine receptor A2a
AA2BR_HUMAN           107      P29275  Adenosine receptor A2b
AA3R_HUMAN            127      P33765  Adenosine receptor A3
ACM1_HUMAN            500      P11229  Muscarinic acetylcholine receptor M1
```

出力される表中の protein ID が結合予測計算時に使用するタンパク質の ID です。この ID と

accession はタンパク質のデータベースである UniProt(http://www.uniprot.org/)で使用されている ID です。また# of compounds の列は DB ファイルに登録されているタンパク質ごとの活性化合物数を表しています。一般的に化合物数が大きいほど予測精度も高くなる傾向があります。

3-2. 化合物スクリーニング（化合物の記述子計算からスコアリングまで）

【化合物の記述子計算】

ターゲットとするタンパク質に対しての化合物の予測計算を始める前に、化合物構造（SD ファイル）から記述子を計算する必要があります。化合物記述子の種類は必ず DB ファイル中のものと一致させなければなりません。さらに記述子計算時の化合物の処理条件（脱塩や電荷中性化など）も一致させる必要があります。CzeekS にサンプルとして添付しているファイルの記述子は DRAGON6 をディレクトリ exec 下のスクリプトファイルを用いて計算しており、化合物は脱塩・電荷中性化を行っております。

SMILES ファイルから DRAGON6 で記述子を計算する場合は下の様なコマンドで実行できます。このコマンドでは計算された記述子が標準出力されます。SMILES ファイルを作成する場合は OpenBabel を使って SD ファイルから SMILES ファイルに変換します。

```
$ babel -isdf sample_mols.sdf -osmi sample_mols.smi⏎  SMILES ファイルが無い場合に実行します
$ calc_dragon.sh sample_mols.smi > output.csv⏎
$ cat output.csv⏎
ZINC00074638,315.320,8.522,24.952,38.109,25.091,…
ZINC00075927,269.300,8.416,21.796,32.563,22.216,…
ZINC00492910,300.390,7.152,25.928,42.138,27.228,…
ZINC02759964,339.170,10.941,21.362,32.153,21.784,…
ZINC03518134,264.360,6.778,22.928,39.138,24.228,…
……
フォーマットはコンマ区切り（CSV）になります
```

記述子ファイルのフォーマットはコンマ区切りで、“（化合物 ID），（記述子 1），（記述子 2），…”のように 1 行に 1 化合物ずつ、化合物 ID と記述子の数値を並べて記述して下さい。特に calc_dragon.sh スクリプトを使わず計算する場合はフォーマットに注意して下さい。

【スコアリング】

化合物の記述子が用意できれば、”cgbvs predict”コマンドで予測計算が実行できます。CzeekS にはサンプル記述子ファイルとして”sample_mols.csv”があり、上述のコマンドで実行した内容と同一です。例えば、アドレナリンβ2 受容体に対するスコアの計算は以下のようなコマンドで実行でき、結果は標準出力されます。

```
$ cgbvs predict gpcr_sample.db ADRB2_HUMAN sample_mols.csv⏎
compound        ADRB2_HUMAN
ZINC00074638    0.28438710
ZINC00075927    0.20908271
ZINC00492910    0.93351328
ZINC02759964    0.21094714
ZINC03518134    0.23423021
ZINC03912658    0.21199297
ZINC04143221    0.21139321
……
```

このコマンドの引数 2 は CGBVS モデルの DB ファイルを指定します。引数 3 はターゲットのタンパク ID を指定して、最後の引数 4 に化合物記述子のファイル名を指定します。ここで引数 3 にて指定できるタンパク ID は前述の"cgbvs status –p"コマンドでご確認ください。尚、計算結果をファイルにする場合はリダイレクトして下さい。

【複数タンパクについてスコアリング】

ターゲットタンパクを指定する引数 3 では、コンマ区切りで複数のタンパク ID を並べて指定することによって複数のタンパクについてのスコアを計算することができます。タンパク ID 数には特に上限は設けておりません。例えばβ1、β2 受容体の両方のスコアを計算したい場合は以下のようにコマンドを実行します。

```
$ cgbvs predict gpcr_sample.db ADRB1_HUMAN,ADRB2_HUMAN sample_mols.csv⏎
compound        ADRB1_HUMAN     ADRB2_HUMAN
ZINC00074638    0.17976740      0.28438710
ZINC00075927    0.20880457      0.20908271
ZINC00492910    0.95092929      0.93351328
ZINC02759964    0.21085676      0.21094714
ZINC03518134    0.21404927      0.23423021
ZINC03912658    0.21199363      0.21199297
ZINC04143221    0.20920101      0.21139321
……
```

この様に複数タンパクについてのスコアがタブ区切りで表示されます。複数タンパクを指定すれば、化合物の選択性も考慮したスクリーニングも容易です。また、%記号をワイルドカードとして利用することもできます。例えば、α受容体も含めた全てのアドレナリン受容体に対してのスクリーニングは以下のようなコマンドで実行できます。

```
$ cgbvs predict gpcr_sample.db ADA%,ADR% sample_mols.csv⏎
compound        ADA1A_HUMAN     ADA1B_HUMAN     ADA1D_HUMAN     ADA2A_HUMAN
ADA2B_HUMAN     ADA2C_HUMAN     ADRB1_HUMAN     ADRB2_HUMAN     ADRB3_HUMAN
ZINC00074638    0.12540752      0.12634313      0.13616720      0.17850074
0.18473179      0.16914742      0.17976740      0.28438710      0.15973684
ZINC00075927    0.20679829      0.20830547      0.20429273      0.20926718
0.20957177      0.21093682      0.20880457      0.20908271      0.20811030
ZINC00492910    0.65030175      0.56015894      0.45491847      0.13154727
0.17594536      0.28564372      0.95092929      0.93351328      0.92305907
……
```

【表示形式】

“cgbvs predict”のオプションで CGBVS スコアの表示内容を変更することができます。-d オプションをつけると正規化されたスコアではなく、SVM の決定関数の平均値を出力できます。

```
$ cgbvs predict -d gpcr_sample.db ADR% sample_mols.csv⏎
compound        ADRB1_HUMAN     ADRB2_HUMAN     ADRB3_HUMAN
ZINC00074638    -0.26478072     -0.20163076     -0.28012058
ZINC00075927    -0.24468885     -0.24450721     -0.24513671
ZINC00492910    0.21397780      0.17969116      0.16218763
ZINC02759964    -0.24337111     -0.24331003     -0.24326666
ZINC03518134    -0.24204065     -0.23124308     -0.25057034
ZINC03912658    -0.24264403     -0.24264444     -0.24264354
ZINC04143221    -0.24443639     -0.24302721     -0.24526098
……
```

また-v オプションで決定関数値と正規化スコアの両方を出力します。

```
$ cgbvs predict -v gpcr_sample.db ADR% sample_mols.csv⏎
compound        protein probability     score
ZINC00074638    ADRB1_HUMAN     0.17976740      -0.26478072
ZINC00074638    ADRB2_HUMAN     0.28438710      -0.20163076
ZINC00074638    ADRB3_HUMAN     0.15973684      -0.28012058
ZINC00075927    ADRB1_HUMAN     0.20880457      -0.24468885
ZINC00075927    ADRB2_HUMAN     0.20908271      -0.24450721
ZINC00075927    ADRB3_HUMAN     0.20811030      -0.24513671
ZINC00492910    ADRB1_HUMAN     0.95092929      0.21397780
ZINC00492910    ADRB2_HUMAN     0.93351328      0.17969116
ZINC00492910    ADRB3_HUMAN     0.92305907      0.16218763
……
```

このように、1 行に化合物-タンパク質ペアに対する 2 種のスコア値を表示する形式で表示されます。

3-3. ターゲット予測

（CGBVS におけるターゲット予測とは）

前節では CGBVS で複数タンパク質についてスコアを算出できることについて説明しましたが、この考え方を拡張して計算できる全てのタンパク質についてスコアを算出すれば化合物のターゲット探索にも応用することができます。

“cgbvs predict”のターゲットを指定する引数に”all”を指定すれば、DB ファイルに登録されている全てのタンパクについてスコアを計算できます。（計算できるタンパクは cgbvs status –pv で確認できます）たとえばサンプルの” sample_mols.csv”中の ZINC10454282 という ID の化合物について全スコアを計算する例は以下のようになります。

```
$ grep ZINC10454282 sample_mols.csv > test.csv⏎
$ cgbvs predict -v gpcr_sample.db all test.csv⏎
compound        protein probability     score
ZINC10454282    5HT1A_HUMAN     0.20557829      -0.25556842
ZINC10454282    5HT1B_HUMAN     0.22102972      -0.23974343
ZINC10454282    5HT1D_HUMAN     0.22942645      -0.23286491
ZINC10454282    5HT1E_HUMAN     0.55217696      -0.08051948
ZINC10454282    5HT1F_HUMAN     0.26455111      -0.21233710
ZINC10454282    5HT2A_HUMAN     0.27088571      -0.21421637
ZINC10454282    5HT2B_HUMAN     0.31850636      -0.18550090
ZINC10454282    5HT2C_HUMAN     0.22478987      -0.23983607
ZINC10454282    5HT4R_HUMAN     0.21420326      -0.24355161
ZINC10454282    5HT5A_HUMAN     0.39039078      -0.15037319
ZINC10454282    5HT6R_HUMAN     0.26206037      -0.21316128
ZINC10454282    5HT7R_HUMAN     0.21391837      -0.24710130
ZINC10454282    A4_HUMAN        0.20011223      -0.25154650
ZINC10454282    AA1R_HUMAN      0.14379993      -0.29728966
ZINC10454282    AA2AR_HUMAN     0.19187479      -0.25688008
ZINC10454282    AA2BR_HUMAN     0.18312868      -0.26278225
```

この例では、-v オプションを付けて行方向にタンパク ID を並べて表示させています。出力をリダイレクトしてスコア高い順にソートすれば確率の高いターゲットのリストが得られます。

```
$ cgbvs predict -v gpcr_sample.db all test.csv > out⏎
$ sort –k3 –nr out | head⏎
ZINC10454282    MTR1A_HUMAN     0.86156594      0.09716781
ZINC10454282    MTR1B_HUMAN     0.81599677      0.05656762
ZINC10454282    TSHR_HUMAN      0.70727932      -0.00874495
ZINC10454282    GRM2_HUMAN      0.70480139      -0.01006024
ZINC10454282    5HT1E_HUMAN     0.55217696      -0.08051948
ZINC10454282    CCR3_HUMAN      0.50019791      -0.10275949
ZINC10454282    ACM3_HUMAN      0.41991731      -0.13739126
ZINC10454282    ACM5_HUMAN      0.40762268      -0.14497484
ZINC10454282    HRH3_HUMAN      0.40188534      -0.14572842
ZINC10454282    5HT5A_HUMAN     0.39039078      -0.15037319
```

先頭 2 行のタンパク ID の MTR1A_HUMAN と MTR1B_HUMAN は以下のようにして確認できます。

```
$ cgbvs status -pv gpcr_sample.db | grep -e "MTR1..*"⏎
MTR1A_HUMAN           102     P48039  Melatonin receptor type 1A
MTR1B_HUMAN           101     P49286  Melatonin receptor type 1B
```

3-4. 構造類似度（Tanimoto 係数）の計算

CzeekS では化合物のフィンガープリントから、Tanimoto 係数（Similarity）を計算することができます。Tanimoto 係数を計算する対象は、ターゲットとして指定したタンパク質と結合する化合物群（DB ファイル中）です。複数の化合物との Tanimoto 係数を計算し、最大値のものを表示します。コマンド操作は"cgbvs predict"に-s オプションを付けて実行します。以下にその手順について示します。

```
$ calc_FP_MACCS sample_mols.sdf test.fp⏎  フィンガープリント計算です。test.fp と sample_mols.fp は同一のものになります
$ cgbvs predict -s gpcr_sample.db ADRB2_HUMAN test.fp
compound        ADRB2_HUMAN
ZINC00074638    0.55737705
ZINC00075927    0.48571429
ZINC00492910    0.71428571
ZINC02759964    0.58108108
ZINC03518134    0.56666667
ZINC03912658    0.72000000
ZINC04143221    0.72972973
ZINC05766699    0.54385965
```

フィンガープリントファイル test.fp の内容は以下のようになります。

```
$ head sample_mols.fp⏎
ZINC00074638,42 50 57 62 72 75 76 83 85 87 89 91 92 95…
ZINC00075927,41 42 52 65 75 78 80 87 92 94 95 97 98 107 110…
ZINC00492910,54 72 82 90 92 95 97 100 104 109 110 113 117 126…
ZINC02759964,24 46 49 52 56 63 65 70 71 75 79 80 83 87 92 93…
ZINC03518134,65 72 75 83 85 90 91 92 93 95 96 104 110 111 117…
…
```

書式は、1 列目に化合物 ID を記述してコンマで区切り、2 列目にフィンガープリントの内容を記述します。フィンガープリント部分の書式は、1 が立っているビットのビット番号（*n* 桁ビット列の左端を最下位 1 とし、右端を最上位 *n* とする）をスペース区切りで記述してください。

4. CGBVS モデルの作成と独自データの追加

4-1. モデル作成に必要なデータとフォーマット

CGBVS の学習モデルの作成に必要な情報は以下の 3 点です。

1. 化合物の記述子情報
2. タンパクの記述子情報
3. 化合物-タンパク質ペアの相互作用情報

上記 3 つの情報をコンマ区切り（CSV）のファイルとして用意しなければなりません。ファイルの書式についてモデル作成用のサンプルデータを例に説明します。

先ずは化合物の記述子情報についてです。サンプルの training_mols.csv を見てみます。

```
$ head training_mols.csv⏎
1000029,419.62,6.557,38.396,63.214,41.347,72.142,0.6,0.988,0.646,…
1000123,279.35,8.73,21.03,32.782,21.835,36.119,0.657,1.024,0.682,…
100014,377.35,8.029,30.009,46.891,32.353,53.033,0.638,0.998,0.688,…
1000194,405.5,7.651,33.993,53.443,35.245,59.857,0.641,1.008,0.665,…
1000948,246.24,8.794,19.009,29.047,18.875,31.495,0.679,1.037,0.674,…
1000956,399.54,9.08,30.072,44.618,31.801,49.242,0.683,1.014,0.723,…
1001098,216.32,6.76,19.246,31.709,20.591,36.484,0.601,0.991,0.643,…
1001421,300.51,8.839,22.007,33.945,24.739,37.872,0.647,0.998,0.728,…
100163,481.66,6.784,42.746,70.829,45.466,80.149,0.602,0.998,0.64,…
1001651,336.37,8.204,27.59,41.698,28.159,45.741,0.673,1.017,0.687,…
```

3 章で化合物のスコアリングの時に用いた記述子ファイルと同一のフォーマットになります。第 1 列目に化合物 ID を記述して、続く 2 列目以降に数値を記述していく形になります。この例は training_mols.smi から DRAGON6 を用いて計算した結果です。

次にタンパク質の記述子情報についてです。タンパク質についても化合物と同様の書式になります。gpcr.csv というファイル名でサンプルを用意してあります。

```
$ head gpcr.csv⏎
5HT1A_HUMAN,9.71564,3.317536,3.791469,3.554502,4.028436,…
5HT1B_HUMAN,8.974359,2.820513,3.589744,3.333333,4.358974,…
5HT1D_HUMAN,9.814324,2.917772,2.65252,3.183024,4.509284,…
5HT1E_HUMAN,6.575342,3.287671,3.561644,3.287671,4.657534,…
5HT1F_HUMAN,6.284153,3.005464,4.098361,4.644809,4.371585,…
5HT2A_HUMAN,6.157113,3.184713,4.246285,3.821656,5.307856,…
5HT2B_HUMAN,6.029106,1.663202,2.910603,4.365904,5.405405,…
5HT2C_HUMAN,5.895197,2.620087,2.838428,4.803493,4.585153,…
5HT4R_HUMAN,6.958763,4.639175,3.865979,3.092784,5.670103,…
5HT5A_HUMAN,7.843137,2.80112,2.521008,3.921569,6.162465,…
```

この例は PROFEAT(http://bidd.cz3.nus.edu.sg/cgi-bin/prof/protein/profnew.cgi)のサービスを使って FASTA ファイルから計算した例です。計算方法などの詳しい情報は PROFEAT のページを参照して下さい。CzeekS では、タンパク質の ID として UniProt の ID を採用しておりますので、特殊なタンパク質でない場合はなるべく「*_HUMAN」で表現される UniProtID を用いて下さい。

最後に相互作用情報についてです。サンプルの positive.csv というファイルを見てみます。

```
$ head positive.csv⏎
1000029,NPBW1_HUMAN
1000123,ARBK1_HUMAN
100014,CRFR1_HUMAN
1000194,FAK2_HUMAN
1000948,CCR6_HUMAN
1000956,NTR1_HUMAN
1001098,FAK2_HUMAN
1001421,OX1R_HUMAN
100163,PTAFR_HUMAN
1001651,ADRB2_HUMAN
```

このファイルの書式は、第１列に化合物 ID を記述し、第２列にタンパク質 ID を記述します。このように化合物-タンパク質のペアを行方向に記述していきます。この例では ChEMBL データベースにて、化合物-タンパク質の組み合わせで活性値が 30μM 以下のものからサンプリングしております。

4-2. モデルファイル（DB ファイル）の作成

前述したモデル作成に必要な３つのファイルが用意できれば CGBVS のモデルファイル(DB ファイル)を作成することができます。ここではサンプルファイル（training_mols.csv, gpcr.csv, positive.csv）を利用した例を紹介します。以下のようにコマンド操作して下さい。

```
$ cgbvs create training.db⏎ 空の DB ファイルの作成
$ cgbvs import training.db training_mols.csv compound⏎ 化合物記述子の登録
import training_mols.csv
$ cgbvs import training.db gpcr.csv protein⏎ タンパク質記述子の登録
import gpcr.csv
$ cgbvs import training.db positive.csv positive⏎ 相互作用情報の登録
import positive.csv
```

先ず、空の DB ファイルを作成します。その後、３つのファイルを DB ファイルにインポート（順不同）して下さい。尚、”cgbvs create”にてオプション指定することにより、DB ファイルの作成と同時にファイルをインポートできます。ここまでの段階で SVM を用いて機械学習を行い、CGBVS モデルを構築することができます。機械学習については 4-4 を参照して下さい。

3-4 で説明したように、CzeekS では DB ファイルに登録してある化合物との構造類似度（Tanimoto 係数）を計算することができます。構造類似度を計算する際には、登録した化合物記述子と同化合物のフィンガープリントが登録されていなければなりません。フィンガープリント登録の操作は以下のようなコマンドとなります。

```
$ cgbvs import training.db training_mols.fp fingerprint⏎
import training_mols.fp
```

フィンガープリントファイルの書式や、MACCS の計算方法については 3-4 を参照して下さい。

4-3. データの追加

この節では既に存在する DB ファイルに別途データ（独自のアッセイデータ）を追加して、CGBVS モデルを更新する方法について説明します。用意すべき情報は基本的には 4-1 で述べた 3 種類です。尚、タンパク質の記述子情報については既に登録済みのものについては用意する必要はありません。登録されているかどうかを確認するには"cgbvs status"コマンドに-pv オプションを付けて実行して下さい。-pv はリガンド数 0 のタンパク質も表示するオプションです。詳しくは 3-1 を参照して下さい。

データの追加方法は"cgbvs add"コマンドを使います。サンプルデータとしてヒスタミン H3 受容体の化合物 100 個を H3_mols.sdf というファイルで用意してあります。これらを記述子計算したファイルが H3_mols.csv です。また、相互作用情報ファイルは H3_positive.csv となります。タンパク質の記述子は既に登録済みなので、追加の必要はありません。

```
$ cgbvs add training.db H3_mols.csv compound⏎  化合物記述子の追加
import H3_mols.csv
$ cgbvs add training.db H3_positive.csv positive⏎  相互作用情報の追加
import H3_positive.csv
```

4-4. 機械学習

DB ファイルにデータを登録あるいは追加した後に SVM による機械学習を行う必要があります。機械学習は"cgbvs learn"コマンドで以下のように実行することができます。

```
$ cgbvs learn -c 10 -g 0.01 training.db 5⏎
output input_1
SVMlearn -c 10.000000 -g 0.010000 -v 5 input_1 model_1⏎
    itr       nSV      vKKT                Objective
      1       978     42378    -4.497671328644441E+02
      2      1907     41404    -8.200883693534472E+02
      3      2786     43240    -1.321260914509097E+03
```

上記の例は負例を 5 セット作成する場合で、負例セット数は最後の引数で指定します。この値は通常 5～10 程度を指定します。負例セットについては 3-1 を参照して下さい。また、オプションの-c と-g は SVM のパラメータです。-c で SVM のソフトマージンに関するパラメータ C を指定します。CzeekS では SVM のカーネル関数としてガウス型の RBF(Radial Basis Function)関数を用いております。-g は RBF 関数のγの値を指定します。

上記コマンド例では C=10, γ=0.01 として機械学習を実行していますが、SVM パラメータの値によって予測精度が変化します。幾つかの C とγの組み合わせを探索して最適な値を設定してください。パラメータ探索の一例は次節で解説します。

4-5. その他

4-4 では機械学習の実行方法について解説しました。5 セットの負例を作って計算する場合を例にコマ

ンド操作を示しましたが、この例では 5 回の機械学習を直列に実行します。複数の計算機を有する場合は、これらの負例セットについて並列に計算することも可能です。ここでは、負例セットごとに独立に（並列に）機械学習計算する方法についてコマンド操作を紹介します。まずは以下のように-f オプションを付けて SVM へのインプットファイルを作成のみを行います。

```
$ cgbvs learn -f training.db 5⏎
output input_1
output input_2
output input_3
output input_4
output input_5
```

次に、それぞれの計算機にて以下のように SVM の機械学習を実行します。

```
$ SVMlearn -c 10 -g 0.01 input_1 model_1⏎  計算機 1 で実行
$ SVMlearn -c 10 -g 0.01 input_2 model_2⏎  計算機 2 で実行
$ SVMlearn -c 10 -g 0.01 input_3 model_3⏎  計算機 3 で実行
$ SVMlearn -c 10 -g 0.01 input_4 model_4⏎  計算機 4 で実行
$ SVMlearn -c 10 -g 0.01 input_5 model_5⏎  計算機 5 で実行
```

上記コマンドが正常終了すれば、model_1～model_5 までの 5 つのファイルが作成されます。これらを以下のコマンドにて DB ファイルに取り込んで下さい。

```
$ cgbvs add_model training.db model_1 1⏎  model_1 を id=1 として取り込む
$ cgbvs add_model training.db model_2 2⏎  model_2 を id=2 として取り込む
$ cgbvs add_model training.db model_3 3⏎  model_3 を id=3 として取り込む
$ cgbvs add_model training.db model_4 4⏎  model_4 を id=4 として取り込む
$ cgbvs add_model training.db model_5 5⏎  model_5 を id=5 として取り込む
```

取り込んだモデルを確認する場合は”cgbvs status”コマンドで確認できます。

最適パラメータを探索する場合も上記の要領で実行できます。以下のスクリプトは、input_1 のみを利用してパラメータ探索する例です。

```
#!/bin/sh

for c in 1 3 10 30 100; do
 for g in 0.001 0.003 0.01 0.03 0.1; do
  echo -ne $c"¥t"$g"¥t"
  SVMlearn -c $c -g $g input_1 model_1 | grep cross-validation | awk '{print $6}'
 done
done
```

SVM パラメータの C=1, 3, 10, 30, 100 の 5 通りで、γ=0.001, 0.003, 0.01, 0.03, 0.1 の 5 通りの組み合わせの計算を実行します。出力形式は C、γ、予測率の順で表示されます。最も予測率の高い C とγの組み合わせで model_1～model_5 を計算して DB ファイルに取り込んで下さい。

5. cgbvs コマンドレファレンス

【使用方法】

```
cgbvs <サブコマンド> [<オプション>] <引数>
```

利用可能なサブコマンドは次の通りです。

add, add_model, create, delete, import, learn, predict, status

オプションや引数はそれぞれのサブコマンド毎に異なります。

【サブコマンドの説明】

add　データを追加します

（書式）

```
cgbvs add <db ファイル> <データファイル> <ターゲット>
```

（説明）

記述子情報や相互作用ペア情報などのデータファイル（CSV 形式）を db ファイルに取り込み現存のデータに追加します。また、引数の<ターゲット>はデータファイルの種類（化合物の記述子情報や相互作用ペア情報など）を指定します。指定できるターゲットの種類は次の通りです。

compound	化合物の記述子
protein	タンパクの記述子
positive	正例の相互作用ペア
negative	負例の相互作用ペア
fingerprint	化合物のフィンガープリント

add_model　SVM のモデルファイルを追加します

（書式）

```
cgbvs add_model [オプション] <db ファイル> <モデルファイル> <ID 番号>
```

（説明）

SVM の機械学習で出力されるモデルファイルを、ID 番号を付与して db ファイルに取り込みます。ここで指定する ID 番号は、プログラムにより生成される負例セットの識別に用いられます。また、現存するモデルの ID 番号を同じものを指定した場合は上書きされてしまうので注意してください。デフォルトでは SVMlearn コマンドにより学習され出力されるモデルファイルを取り込みます。-l をつけると libsvm の svm-train コマンドにより学習されたモデルファイルを取り込みます。

（オプション）

-l	:libsvm のモデルファイルを取り込みます

create　空の db ファイルを作成します

（書式）

cgbvs create [オプション] <db ファイル>

（説明）

何もデータを登録していない db ファイルを作ります。オプションでファイルを指定することにより、db ファイル作成と同時に記述子情報などのデータを登録することができます。ここでオプション指定無しでも、後で import サブコマンドによりデータ登録することができます。

（オプション）

-c <arg>　:<arg>で指定したファイルから化合物記述子を登録します
-p <arg>　:<arg>で指定したファイルからタンパク記述子を登録します
-i <arg>　:<arg>で指定したファイルから正例の相互作用ペアを登録します
-n <arg>　:<arg>で指定したファイルから負例の相互作用ペアを登録します
-f <arg>　:<arg>で指定したファイルから化合物フィンガープリントを登録します

<arg>で指定するファイルは CSV 形式

delete　データを削除します

（書式）

cgbvs delete <db ファイル> <ターゲット>

（説明）

<db ファイル>で指定した db ファイルから、<ターゲット>で指定した種類の情報を削除します。指定できるターゲットの種類は次の通りです。

compound	化合物の記述子
protein	タンパクの記述子
positive	正例の相互作用ペア
negative	負例の相互作用ペア
fingerprint	化合物のフィンガープリント

import　現在のデータを一旦削除して、新たなデータを db ファイルに登録します

（書式）

cgbvs import <db ファイル> <データファイル> <ターゲット>

（説明）

記述子情報や相互作用ペア情報などのデータファイル（CSV 形式）を db ファイルに取り込み登録します。引数の<ターゲット>はデータファイルの種類（化合物の記述子情報や相互作用ペア情報など）を指定します。指定できるターゲットの種類は次の通りです。

compound	化合物の記述子
protein	タンパクの記述子

positive　　正例の相互作用ペア
negative　　負例の相互作用ペア
fingerprint　化合物のフィンガープリント

サブコマンド add と異なる点は、db ファイル内に現存する<ターゲット>で指定されたデータを一旦削除することです。db ファイルに登録されている記述子とは異なる記述子（ベクトル次元が異なるなど）を登録する場合はこの import を利用してください。

learn　機械学習若しくは機械学習のための入力ファイル作成を行います。

（書式）

cgbvs learn [オプション] <db ファイル> <負例セット数>

（説明）

db ファイル内に登録されているデータ（化合物記述子、タンパク記述子、正例の相互作用ペア）を用いて負例セットを生成（ランダムペア）してから、SVM による機械学習を行います。さらに続けて計算されたモデルファイルを db ファイル内に取り込みます。このとき、<負例セット数>で指定された数だけ負例セットが生成されますので、同数の SVM の機械学習が実行されます。もし、複数の負例セットの機械学習を複数台の計算機に分けて行いたい場合は次の様な手順で行ってください。先ず、オプション指定で一旦 SVM の入力ファイルを出力します。<負例セット数>で指定した数のファイルが出力されますので、それぞれの計算機で SVM 機械学習を実行の後、モデルファイルを手動で db ファイルに取り込んで下さい。

（オプション）

-c <arg>　　:SVM のソフトマージンのパラメータ C を指定する（初期設定 10）
-g <arg>　　:RBF カーネルのパラメータγを指定する（初期設定 0.01）
-v <arg>　　:交差検定の回数を指定する　（初期設定 5）
-s <arg>　　:1 タンパクあたり化合物数の上限を指定してデータサンプリングする
-pc <arg>　:化合物記述子に対して主成分分析を行い、情報圧縮する
-pp <arg>　:タンパク記述子に対して主成分分析を行い、情報圧縮する

上記 2 オプションの<arg>は整数値の場合は、サンプリングする主成分数を示し、<arg>がパーセンテージ（数値%）の場合は累積寄与率が指定の値になるまで主成分をサンプリングする。

-m　　:負例セット生成は行いません
-n　　:登録された負例セットを使用します
-r　　:負例セットを変更せずに機械学習します

以下の 2 オプションを指定した場合はファイルの出力のみ行い、SVM の機械学習は行いません。

-f　　:SVMlearn コマンドで使用する入力ファイルを出力します
-fl　　:LIBSVM で使用する入力ファイルを出力します

predict　予測スコアを算出します

（書式）

```
cgbvs predict [オプション] <db ファイル> <タンパク ID> <化合物記述子ファイル>
```

（説明）

<db ファイル>で指定した CGBVS モデルを用いて、<タンパク ID>で指定したターゲットに対して<化合物記述子ファイル>で指定したファイル中の化合物の予測スコアを算出します。予測したい化合物は予め記述子を計算して所定の形式でファイルする必要があります。化合物数に上限は設けておりません。また<タンパク ID>はコンマ区切りで複数指定することが可能です。また、'%'を文字列のワイルドカードとして利用でき、"all"を指定とすることで db ファイルに登録されている全てのタンパクについてスコアを算出します。尚、利用可能なタンパク ID 一覧についてはサブコマンド status に-p オプションを付けて確認して下さい。

（オプション）

-s	:指定タンパクの既知化合物群との類似度（Tanimoto 係数）を計算します
-d	:SVM の決定関数の値を表示します
-v	:結合の予測スコアと決定関数値の両方を出力します
-n <arg>	:<arg>で指定したモデル ID のみを利用してスコアを算出します

status　db ファイル中のモデルの状況を表示します

（書式）

```
cgbvs status [オプション] <db ファイル>
```

（説明）

db ファイルに登録されているモデルや相互作用データについての内容を、一覧表として出力します。オプション指定無しの場合は、モデルについての情報が出力されます。

（オプション）

-c	:化合物 ID リストと相互作用するタンパク数が出力されます
-p	:タンパク ID リストと相互作用する化合物数が出力されます
-pv	:全てのタンパク ID リストと相互作用する化合物数が出力されます

-p オプションについては、化合物数が 1 以上のタンパクに限り、化合物数とそのタンパク名を確認できます。-pv オプションでは、登録されている全てのタンパクについて確認できます。サブコマンド predict で利用できるタンパク ID は、このコマンドで一覧表示されるタンパク ID のみです。