

産業技術総合研究所 創薬分子プロファイリング研究センター

# セマンティックWEB 解析サービス

ユーザーマニュアル

AIST

2015/07/24

1.	Sync 同期型解析サービスについて .....	1
1.0.	Sync 同期型解析サービスの利用方法 .....	1
1.1.	Blast .....	5
1.1.1.	入力 RDF の準備 .....	5
1.1.2.	実行コマンド .....	6
1.1.3.	実行結果 .....	7
1.2.	CentroidFold.....	9
1.2.1.	入力 RDF の準備 .....	9
1.2.2.	実行コマンド .....	10
1.2.3.	実行結果 .....	11
1.3.	ClustalW.....	13
1.3.1.	入力 RDF の準備 .....	13
1.3.2.	実行コマンド .....	14
1.3.3.	実行結果 .....	15
1.4.	IPknot.....	17
1.4.1.	入力 RDF の準備 .....	17
1.4.2.	実行コマンド .....	18
1.4.3.	実行結果 .....	19
1.5.	Mafft.....	20
1.5.1.	入力 RDF の準備 .....	20
1.5.2.	実行コマンド .....	22
1.5.3.	実行結果 .....	22
1.6.	Psipred.....	24
1.6.1.	入力 RDF の準備 .....	24
1.6.2.	実行コマンド .....	25
1.6.3.	実行結果 .....	26
1.7.	Raccess .....	28
1.7.1.	入力 RDF の準備 .....	28
1.7.2.	実行コマンド .....	29
1.7.3.	実行結果 .....	30
1.8.	RactIP.....	32
1.8.1.	入力 RDF の準備 .....	32
1.8.2.	実行コマンド .....	33
1.8.3.	実行結果 .....	34
1.9.	Wolfpsort .....	36
1.9.1.	入力 RDF の準備 .....	36

1.9.2.	実行コマンド .....	37
1.9.3.	実行結果 .....	38
2.	Async 非同期型解析サービスについて .....	39
2.0.	Async 同期型解析サービスの利用方法 .....	39
2.1.	Last .....	41
2.1.1.	入力 RDF の準備 .....	41
2.1.2.	実行コマンド .....	43
2.1.3.	実行結果 .....	44
2.2.	Modelling .....	46
2.2.1.	入力 RDF の準備 .....	46
2.2.2.	実行コマンド .....	50
2.2.3.	実行結果 .....	52
2.3.	PoodleL .....	54
2.3.1.	入力 RDF の準備 .....	54
2.3.2.	実行コマンド .....	55
2.3.3.	実行結果 .....	56
2.4.	PoodleS .....	57
2.4.1.	入力 RDF の準備 .....	57
2.4.2.	実行コマンド .....	58
2.4.3.	実行結果 .....	60
2.5.	Rassie .....	61
2.5.1.	入力 RDF の準備 .....	61
2.5.2.	実行コマンド .....	63
2.5.3.	実行結果 .....	65
おわりに	.....	67

解析サービス名		入力 RDF を定義した OWL クラス
Blast	S	<a href="http://www.molprof.jp/ontologies/aistlssio.owl#BlastInput">http://www.molprof.jp/ontologies/aistlssio.owl#BlastInput</a>
CentroidFold	S	<a href="http://www.molprof.jp/ontologies/aistlssio.owl#CentroidFoldInput">http://www.molprof.jp/ontologies/aistlssio.owl#CentroidFoldInput</a>
ClustalW	S	<a href="http://www.molprof.jp/ontologies/aistlssio.owl#ClustalWInput">http://www.molprof.jp/ontologies/aistlssio.owl#ClustalWInput</a>
IPknot	S	<a href="http://www.molprof.jp/ontologies/aistlssio.owl#IPknotInput">http://www.molprof.jp/ontologies/aistlssio.owl#IPknotInput</a>
Mafft	S	<a href="http://www.molprof.jp/ontologies/aistlssio.owl#MafftInput">http://www.molprof.jp/ontologies/aistlssio.owl#MafftInput</a>
Psipred	S	<a href="http://www.molprof.jp/ontologies/aistlssio.owl#PsiPredInput">http://www.molprof.jp/ontologies/aistlssio.owl#PsiPredInput</a>
Raccess	S	<a href="http://www.molprof.jp/ontologies/aistlssio.owl#RaccessInput">http://www.molprof.jp/ontologies/aistlssio.owl#RaccessInput</a>
RactIP	S	<a href="http://www.molprof.jp/ontologies/aistlssio.owl#RactIPInput">http://www.molprof.jp/ontologies/aistlssio.owl#RactIPInput</a>
Wolfpsort	S	<a href="http://www.molprof.jp/ontologies/aistlssio.owl#WolfPsortInput">http://www.molprof.jp/ontologies/aistlssio.owl#WolfPsortInput</a>
Last	A	<a href="http://www.molprof.jp/ontologies/aistlssio.owl#LastInput">http://www.molprof.jp/ontologies/aistlssio.owl#LastInput</a>
Modelling	A	<a href="http://www.molprof.jp/ontologies/aistlssio.owl#ModellingInput">http://www.molprof.jp/ontologies/aistlssio.owl#ModellingInput</a>
PoodleL	A	<a href="http://www.molprof.jp/ontologies/aistlssio.owl#PoodleLInput">http://www.molprof.jp/ontologies/aistlssio.owl#PoodleLInput</a>
PoodleS	A	<a href="http://www.molprof.jp/ontologies/aistlssio.owl#PoodleSInput">http://www.molprof.jp/ontologies/aistlssio.owl#PoodleSInput</a>
Rassie	A	<a href="http://www.molprof.jp/ontologies/aistlssio.owl#RassieInput">http://www.molprof.jp/ontologies/aistlssio.owl#RassieInput</a>

表 1-A 入力となる RDF を定義した OWL クラス： S (同期型), A (非同期型)

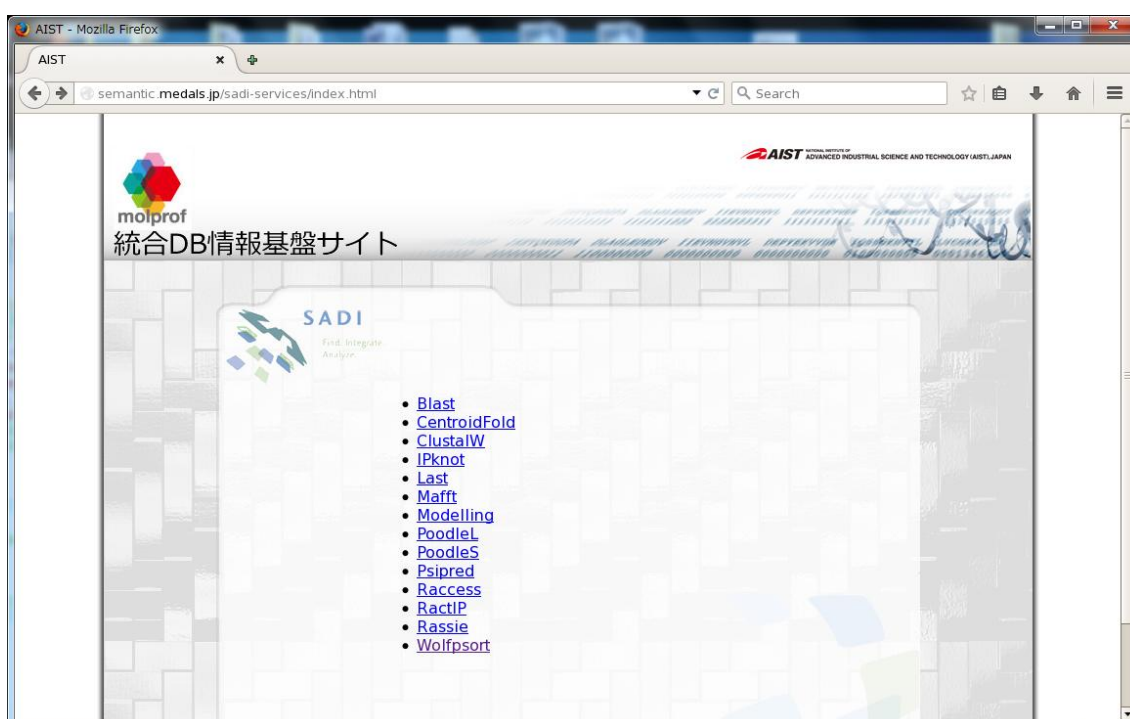
## 1. Sync 同期型解析サービスについて

---

### 1.0. Sync 同期型解析サービスの利用方法

---

まず、<http://semantic.medals.jp:8090/sadi-services/index.html> にアクセスすると、利用可能な SADI の解析サービス一覧が表示される。



SADI サービス

このうち、Sync 同期型解析サービス(Blast, CentroidFold, ClustalW, IPknot, Mafft, Psipred, Raccess, RactIP, Wolfpsort)について、以下の cURL コマンドを用いて各解析サービスを実行する。

```
% curl --data-binary @入力 RDF 名
http://semantic.medals.jp:8090/sadi-services/解析サービス名 (表 1-A 参照) -o 出力 RDF 名
```

例えば、input.rdf を入力 RDF として用意し、Wolfpsort を実行、その実行結果を output.rdf という RDF で出力する場合、以下のコマンドとなる。

```
% curl --data-binary @input.rdf
http://semantic.medals.jp:8090/sadi-services/Wolfpsort -o output.rdf
```

用意する入力 RDF のフォーマットは以下の通りである。

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#"
  <aistls:WolfPsortInput
    rdf:about="http://www.molprof.jp/ontologies/wolfpsort.rdf#1">
      <aistls:requiresKingdomInformationOf>animal</aistls:requiresKingdomInformationOf>
      <aistls:requiresQueryProteinSequence>
        MANLGCWMLVLFVATWSDLGLCKKRPKPGGWNTGGSRYPGQSPGGNRYPPQGGGGWGQPHGGGGWGQPHGGGGWGQPHGGGGWG
        QPHGGGGWGQGGGTHSQWNKPSKPKTNMKHMAGAAAAGAVVGLGGYMLGSAMSRPIIHFGSDYEDRYRENMHRYPNQVYYRPMDEYSN
      </aistls:requiresQueryProteinSequence>
    </aistls:WolfPsortInput>
</rdf:RDF>
```

### Wolfpsort 用入力 RDF

- ・ 黒文字部分  
→RDF の開始終了タグ(各解析サービス共通)
- ・ 緑文字部分  
→名前空間指定(各解析サービス共通)
- ・ 赤文字部分

→入力 RDF を定義した OWL クラス(表 1-A 参照)及び URI(任意)を指定  
<Wolfpsort の場合>

入力 RDF を定義した OWL クラス : aistls:WolfPsortInput

URI(任意) : http://www.molprof.jp/ontologies/wolfpsort.rdf#1

・ 青文字部分

→各解析サービス実行に必要なトリプルを指定

<Wolfpsort の場合(指定するトリプルは2つ)>

述語

目的語

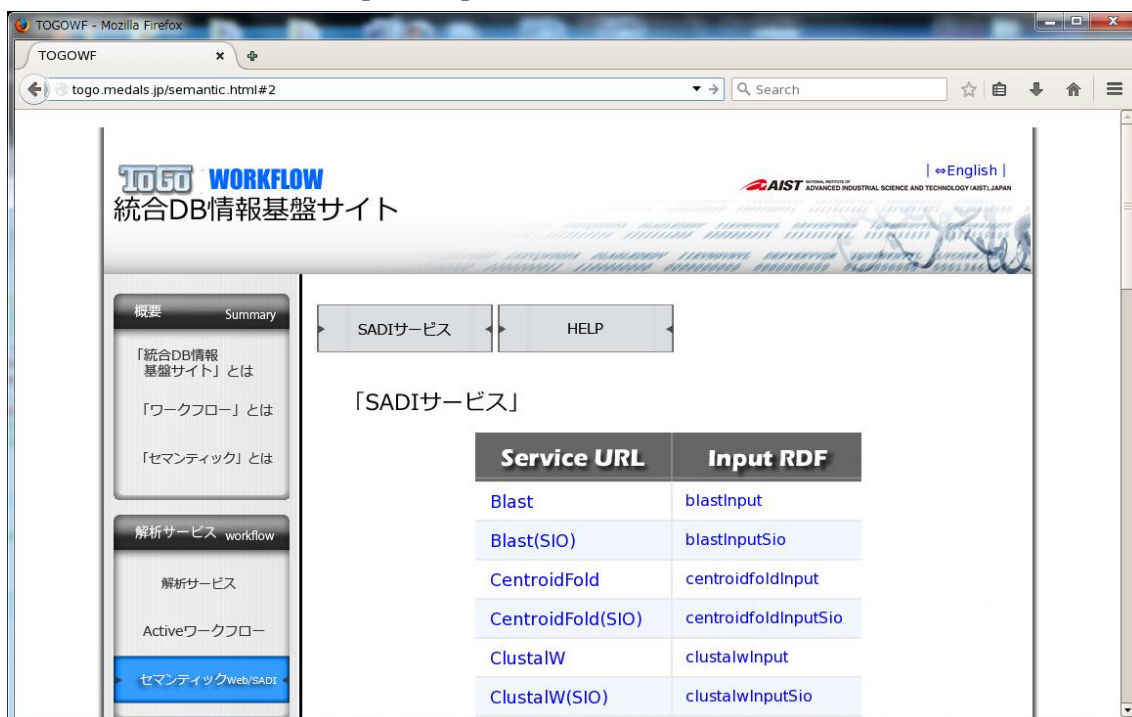
requiresKingdomInformationOf

界情報(animal, plant, fungi から一つ選択)

requiresQueryProteinSequence

アミノ酸配列

なお、入力用 RDF のサンプルファイルを入手することができる。SADI サービスページ(<http://togo.medals.jp/semantic.html#2>)にアクセスし、Wolfpsort の入力用 RDF を入手する場合であれば wolfpsortInput をクリックする。



The screenshot shows a web browser window displaying the SADI service page. The page title is "TOGO WORKFLOW 統合DB情報基盤サイト". The main content area is titled "「SADIサービス」" and contains a table with two columns: "Service URL" and "Input RDF". The table lists several services and their corresponding input RDF files.

Service URL	Input RDF
Blast	blastInput
Blast(SIO)	blastInputSio
CentroidFold	centroidfoldInput
CentroidFold(SIO)	centroidfoldInputSio
ClustalW	clustalwInput
ClustalW(SIO)	clustalwInputSio

SADI サービスページ

実行結果が格納される出力 RDF のフォーマットは以下の通りである。

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#">
  <aistls:WolfPsortOutput rdf:about="http://www.molprof.jp/ontologies/wolfpsort.rdf#1">
    <aistls:requiresResultInTextFormat># k used for kNN is: 32
    queryProtein extr 19, E.R. 4, pero 4, lyso 3, E.R._mito 3, mito_pero 3
    </aistls:requiresResultInTextFormat>
  </aistls:WolfPsortOutput>
</rdf:RDF>
```

### Wolfpsort 用出力 RDF

- ・ 青文字部分

→各解析サービスの実行結果がトリプル形式で格納

<Wolfpsort の場合>

述語

目的語

requiresResultInTextFormat

テキスト形式出力結果

\*cURL をインストールする必要がある際（WINDOWS など）は、下記のサイトよりダウンロード可能です。

<http://curl.haxx.se/>





## 1.1. Blast

---

### 1.1.1. 入力 RDF の準備

---

以下の方法で Blast の入力 RDF を作成する。

- RDF のヘッダに使用語彙 RDF、AISTLS SIO OWL を以下のように定義する。

```
<rdf:RDF
```

```
  xmlns:rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls "http://www.molprof.jp/ontologies/aistlssio.owl#"
```

- 主語は BlastInput クラス、rdf:about 属性は任意とする。

```
<aistls:BlastInput rdf:about="http://www.molprof.jp/ontologies/blast.rdf#1">
```

- 問い合わせ配列のトリプルを定義する。

```
  主語 : BlastInput
```

```
  述語 : requiresQuerySequence
```

```
  目的語 : 文字列(問い合わせ配列)
```

```
<aistls:requiresQuerySequence>問い合わせ配列
```

```
</aistls:requiresQuerySequence>
```

- Blast プログラム名のトリプルを定義する。

```
  主語 : BlastInput
```

```
  述語 : requiresBlastProgramName
```

```
  目的語 : 文字列(blastp, blastn, blastx, tblastn, tblastx)
```

```
<aistls:requiresBlastProgramName>blastp</aistls:requiresBlastProgramName
```

```
>
```

- ターゲットデータベース名のトリプルを定義する。

```
  主語 : BlastInput
```

```
  述語 : requiresBlastDatabase
```

```
  目的語 : 文字列(SWISS, TREMBL, UNIPROT, PROTEIN, PDB etc.)
```

\*その他データベース名は DDBj サイト参照のこと

```
http://blast.ncbi.nlm.nih.gov/Blast.cgi
```

```
<aistls:requiresBlastDatabase>SWISS</aistls:requiresBlastDatabase>
```

- E-value のトリプルを定義する。

```
  主語 : BlastInput
```

述語 : setupEvaluateThreshold

目的語 : 倍精度少数値 (1.0e-10 など)

<aistls:setupEvaluateThreshold>1.0e-30</aistls:setupEvaluateThreshold>

この定義に基づいて Blast の入力 RDF を作成すると以下となる。

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#"
  <aistls:BlastInput rdf:about="http://www.molprof.jp/ontologies/blast.rdf#1">
    <aistls:requiresQuerySequence>>test
MANLGCWMLVLFVATWSDLGLCKKRPKPGGWNTGGSRYPGQSPGGNRYPPQGGGGWGQPHGGGGWGQPHGGGGWGQPHGGGGWGQPHGG
GWQGGGGTHSQWNKPSKPKTNMKHMAGAAAAGAVVGLGGYMLGSAMSRPIIHFGSDYEDRYRENMHRYPNQVYYRPMDEYSNQNN
FVHDCVNIITIKQHTVTTTTTKGENFTETDVKMMERVVEQMCITQYERESQAYYQRSSMVLFSPPVILLISFLIFLIVGMANLGCWML
LVLFVATWSDLGLCKKRPKPGGWNTGGSRYPGQSPGGNRYPPQGGGGWGQPHGGGGWGQPHGGGGWGQPHGGGGWGQPHGGGGWGQGGGT
HSQWNKPSKPKTNMKHMAGAAAAGAVVGLGGYMLGSAMSRPIIHFGSDYEDRYRENMHRYPNQVYYRPMDEYSNQNNFVHDCVNI
TIKQHTVTTTTTKGENFTETDVKMMERVVEQMCITQYERESQAYYQRSSMVLFSPPVILLISFLIFLIVGMANLGCWMLVLFVATW
SDLGLCKKRPKPGGWNTGGSRYPGQSPGGNRYPPQGGGGWGQPHGGGGWGQPHGGGGWGQPHGGGGWGQPHGGGGWGQGGGGTHSQWNKPS
KPKTNMKHMAGAAAAGAVVGLGGYMLGSAMSRPIIHFGSDYEDRYRENMHRYPNQVYYRPMDEYSNQNNFVHDCVNIITIKQHTVT
TTTTTKGENFTETDVKMMERVVEQMCITQYERESQAYYQRSSMVLFSPPVILLISFLIFLIVGMANLGCWMLVLFVATWSDLGLCKK
RPKPGGWNTGGSRYPGQSPGGNRYPPQGGGGWGQPHGGGGWGQPHGGGGWGQPHGGGGWGQPHGGGGWGQPHGGGGWGQGGGGTHSQWNKPS
</aistls:requiresQuerySequence>
    <aistls:requiresBlastDatabase>SWISS</aistls:requiresBlastDatabase>
    <aistls:requiresBlastProgramName>blastp</aistls:requiresBlastProgramName>
    <aistls:setupEvaluateThreshold>1.0e-30</aistls:setupEvaluateThreshold>
  </aistls:BlastInput>
</rdf:RDF>
```

## Blast 用入力 RDF

### 1.1.2. 実行コマンド

利用するマシンから以下のコマンドを実行する。

% curl --data-binary @入力 RDF 名

http://semantic.medals.jp:8090/sadi-services/Blast -o 出力 RDF 名

### 1.1.3. 実行結果

---

以下の定義に従い、Blast の実行結果が RDF 形式で出力される。

- RDF のヘッダは以下となる。

```
<rdf:RDF
```

```
  xmlns:rdf “http://www.w3.org/1999/02/22-rdf-syntax-ns#”
```

```
  xmlns:aistls “http://www.molprof.jp/ontologies/aistlssio.owl#”>
```

- 主語は BlastOutput クラス、rdf:about 属性は入力 RDF ファイルで指定したものがそのまま指定される。

```
<aistls:BlastOutput rdf:about=”http://www.molprof.jp/ontologies/blast.rdf#1”>
```

- Blast のテキスト実行結果のトリプルは以下となる。

```
  主語 : BlastOutput
```

```
  述語 : requiresResultInTextFormat
```

```
  目的語 : 文字列(Blast の実行結果)
```

```
<aistls:requiresResultInTextFormat>Blast 実行結果
```

```
</aistls:requiresResultInTextFormat>
```

Blast の出力結果の例を以下に示す。

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#"
  <aistls:BlastOutput rdf:about="http://www.molprof.jp/ontologies/blast.rdf#1">
    <aistls:requiresResultInTextFormat>&lt;?xml version="1.0"?&gt;
      &lt;!DOCTYPE BlastOutput PUBLIC "-//NCBI//NCBI BlastOutput/EN"
"http://www.ncbi.nlm.nih.gov/dtd/NCBI_BlastOutput.dtd"&gt;
      &lt;BlastOutput&gt;
        &lt;BlastOutput_program&gt;blastp&lt;/BlastOutput_program&gt;
        &lt;BlastOutput_version&gt;BLASTP 2.2.28+&lt;/BlastOutput_version&gt;
        &lt;BlastOutput_reference&gt;Stephen F. Altschul, Thomas L. Madden, Alejandro A.
Sch&amp;amp;auml;ffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997),
&amp;quot;Gapped BLAST and PSI-BLAST: a new generation of protein database search
programs&amp;quot;, Nucleic Acids Res. 25:3389-3402.&lt;/BlastOutput_reference&gt;
        &lt;BlastOutput_db&gt;swissprot&lt;/BlastOutput_db&gt;
        &lt;BlastOutput_query-ID&gt;55685&lt;/BlastOutput_query-ID&gt;
        &lt;BlastOutput_query-def&gt;unnamed protein
product&lt;/BlastOutput_query-def&gt;
        &lt;BlastOutput_query-len&gt;1019&lt;/BlastOutput_query-len&gt;
        &lt;BlastOutput_param&gt;
          &lt;Parameters&gt;
            &lt;Parameters_matrix&gt;BLOSUM62&lt;/Parameters_matrix&gt;
            &lt;Parameters_expect&gt;10&lt;/Parameters_expect&gt;
            &lt;Parameters_gap-open&gt;11&lt;/Parameters_gap-open&gt;
.....
.....
      </aistls:requiresResultInTextFormat>
    </aistls:BlastOutput>
  </rdf:RDF>
```

**Blast 出力結果**

## 1.2. CentroidFold

---

### 1.2.1. 入力 RDF の準備

---

以下の方法で CentroidFold の入力 RDF を作成する。

- RDF のヘッダに使用語彙 RDF、AISTLS SIO OWL を以下のように定義する。

```
<rdf:RDF
```

```
  xmlns:rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls "http://www.molprof.jp/ontologies/aistlssio.owl#"
```

- 主語は CentroidFoldInput クラス、rdf:about 属性を任意に指定する。

```
<aistls:CentroidFoldInput
```

```
  rdf:about="http://www.molprof.jp/ontologies/centroidfold
  .rdf#1">
```

- RNA 配列を入力とする場合、RNA 配列のトリプルを定義する。

主語 : CentroidFoldInput

述語 : requiresQueryRNASequence

目的語 : 文字列(RNA 配列)

```
<aistls:requiresQueryRNASequence>RNA 配列
```

```
</aistls:requiresQueryRNASequence>
```

- ClustalW 形式のマルチプルアラインメントを入力とする場合、ClustalW 形式マルチプルアラインメントのトリプルを定義する。

主語 : CentroidFoldInput

述語 : requiresClustalWMultipleAlignment

目的語 : 文字列(ClustalW 形式マルチプルアラインメント)

```
<aistls:requiresClustalWMultipleAlignment>ClustalW 形式マルチプルアライン
```

```
メント</aistls:requiresClustalWMultipleAlignment>
```

- コマンドラインオプションのトリプルを定義する。

主語 : CentroidFoldInput

述語 : hasOptions

目的語 : 文字列(コマンドラインオプション)

```
<aistls:hasOptions>-g 4</aistls:hasOptions>
```

この定義に基づいて CentroidFold の入力 RDF を作成する。

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#">
  <aistls:CentroidFoldInput
    rdf:about="http://www.molprof.jp/ontologies/centroidfold.rdf#1">
    <aistls:requiresQueryRNASequence>>AF291576.1/215-270 RF00381;Antizyme_FSE;
    UGAUGCCCCUCACCCAUCAGUGAAGAUCGCGGGUGGGCGAGGGAACGGAA
    GGGAUC
    >AAVX01164114.1/511-453 RF00381;Antizyme_FSE;
    UGAUGUCCCUACCCACCCUUGAAGAUCGAGGUGGGCGAGGGAUGGUC
    AAAGGGAUC
    >BAAE01262592.1/2445-2390 RF00381;Antizyme_FSE;
    UGAUGCCCCUCACCCACAGCUGAAGAUCGCGGGUGGGCGAGGGACUGUCA
    GGGAUC
    </aistls:requiresQueryRNASequence>
    <aistls:hasOptions>-g 4</aistls:hasOptions>
  </aistls:CentroidFoldInput>
</rdf:RDF>
```

### CentroidFold 用入力 RDF

#### 1.2.2. 実行コマンド

---

利用するマシンから以下のコマンドを実行する。

```
% curl --data-binary @入力 RDF 名
```

```
http://semantic.medals.jp:8090/sadi-services/CentroidFold -o 出力 RDF 名
```

### 1.2.3. 実行結果

---

以下の定義に従い、CentroidFold の実行結果が RDF 形式で出力される。

- RDF のヘッダは以下となる。

```
<rdf:RDF
```

```
  xmlns:rdf “http://www.w3.org/1999/02/22-rdf-syntax-ns#”
```

```
  xmlns:aistls “http://www.molprof.jp/ontologies/aistlssio.owl#”>
```

- 主語は CentroidFoldOutput クラス、rdf:about 属性は入力 RDF ファイルで指定したものがそのまま指定される。

```
<aistls:CentroidFoldOutput
```

```
  rdf:about=”http://www.molprof.jp/ontologies/centroidfold.rdf#1”>
```

- CentroidFold のテキスト実行結果のトリプルは以下となる。

主語：CentroidFoldOutput

述語：requiresResultInTextFormat

目的語：文字列(CentroidFold の実行結果)

```
<aistls:requiresResultInTextFormat>CentroidFold 実行結果
```

```
</aistls:requiresResultInTextFormat>
```

- CentroidFold の PNG 画像->Base64 変換結果のトリプルは以下となる。

主語：CentroidFoldOutput

述語：requiresResultInBase64BinaryFormat

目的語：文字列(CentroidFold の実行結果)

```
<aistls:requiresResultInBase64BinaryFormat>CentroidFold 実行結果
```

```
</aistls:requiresResultInBase64BinaryFormat>
```

\*入力 RDF ファイルの **hasOptions** に”-noimage”を追加すると、このトリプルは出力されない。

CentroidFold の出力結果の例を以下に示す。

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#">
  <aistls:CentroidFoldOutput
    rdf:about="http://www.molprof.jp/ontologies/centroidfold.rdf#1">
    <aistls:requiresResultInBase64BinaryFormat>PNG:1
      iVBORw0KGgoAAAANSUhEUgAADrcAAA63CAIAAADdRQiTAACAAE1EQVR42uzcW3LtxpFAUU7H32dq
      Cv/0iHoMHhzazSNRFJ8A6pWZtVbgU9dE0b6OSmhHvhwAAAAAAAAAAAAAQc0vfgUAAAAAAAAAAAA
      UIxKGAIAAAAAAAAAAAACqUQkDAAAAAAAAAAAAQDUqYQAAAAAAAAAAAAACoRiUMAAAAAAAAAAAAANWo
      hAEAAAAAAAAAAACGgPuwAAAAAAAAAAAAAFSjEgYAAAAAAAAAAACAalTCAAAAAAAAAAAAAAFcNShgA
      AAAAAAAAAAAAAAQ1EJAwAAAAAAAAAAAAEA1KmEAAAAAAAAAAAAAQEYlDAAAAAAAAAAAAADVqIQBAAAA
      .....
      AACAIvTCcyrhzLS/EmEAAAAAAAAcYwtQNAABgrlQLdluGwvkgpCXCAAAAAAAAAAMJ7BGwAAwFwWCauE
      30iBJcIAAAAAAAAAAwktkbAADdCrhcZVwLYJgfTAAAAAAAAAMyWIHAACwvsJVu41D4YqUwRjHAAAA
      AAAAGMMQDgAAYJG9FwnHV8J16YmLwgAAAAAADCAORwAAMA6KuGoSrg6iTAAAAAAAAAAQzSgOAAABg
      qR9urmevNvTBAAAAAAAAQBwDOeD/t2vHNAAAAiA+rF2NYCbD+QAAIARx9cSboowAAAAAAAAAMBJS
      jJUfYd3SvAAAAABJRU5ErkJggg==
    </aistls:requiresResultInBase64BinaryFormat>
    <aistls:requiresResultInTextFormat>&gt;AF291576.1/215-270 RF00381;Antizyme_FSE;
      UGAUGCCCCUCACCCAUCAGUGAAGAUCCCGGGUGGGCGAGGGAACGGGAAGGGAUC
      ....(.((((.((((.(.....)))))))).)))))..... (g=4,th=0.2,e=-10.39)
      &gt;AAVX01164114.1/511-453 RF00381;Antizyme_FSE;
      UGAUGUCCCUACCCACCCUUGAAGAUCAGGUGGGCGAGGGAUGGUCAAAGGGAUC
      (((.((((.((((.(.....)))))))).)))))..... (g=4,th=0.2,e=-13.76)
      &gt;BAAE01262592.1/2445-2390 RF00381;Antizyme_FSE;
      UGAUGCCCCUCACCCACAGCUGAAGAUCAGGUGGGCGAGGACUGUCAGGGAUC
      (((.((((.((((.(.....)))))))).)))))..... (g=4,th=0.2,e=-12.51)
    </aistls:requiresResultInTextFormat>
  </aistls:CentroidFoldOutput>
</rdf:RDF>
```

### CentroidFold 出力結果



## 1.3. ClustalW

---

### 1.3.1. 入力 RDF の準備

---

以下の方法で ClustalW の入力 RDF を作成する。

- RDF のヘッダに使用語彙 RDF、AISTLS SIO OWL を以下のように定義する。

```
<rdf:RDF
```

```
  xmlns:rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls "http://www.molprof.jp/ontologies/aistlssio.owl#"
```

- 主語は ClustalWInput クラス、rdf:about 属性は任意とする。

```
<aistls:ClustalWInput
```

```
  rdf:about="http://www.molprof.jp/ontologies/clustalw.rdf#1">
```

- 配列のトリプルを定義する。

```
  主語 : ClustalWInput
```

```
  述語 : requiresSequence
```

```
  目的語 : 文字列(核酸またはアミノ酸配列)
```

```
<aistls:requiresSequence>配列(3 つ以上の配列が必
```

```
要)</aistls:requiresSequence>
```

- コマンドラインオプションのトリプルを定義する。

```
  主語 : ClustalWInput
```

```
  述語 : hasOptions
```

```
  目的語 : 文字列(コマンドラインオプション)
```

```
<aistls:hasOptions>-GAPOPEN=10 -GAPEXT=0.5</aistls:hasOptions>
```

この定義に基づいて ClustalW の入力 RDF を作成する。

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#">
  <aistls:ClustalWInput rdf:about="http://www.molprof.jp/ontologies/clustalw.rdf#1">
    <aistls:requiresSequence>>1LYLA
      FNDELNRNRREKLAALRQQGVAFPNDFRRDHTSDQLHEEFDAKDNQELESLNIEVSVAGRMMTRIMGKASFVTLQDVGGR
      I QLYVARDSLPEGVYNDQFKKWDLGDIIIGARGTLFKTQTGELSIHCTELRLLTKALRPLPDQEVRYRQRYLDLIANDKSRQTFVVR
      SK I LAAIRQFMVARGFMEVETPMMQVI PGASARPFITHHNALDLDYMLRIAPELYLKRLVVGGFERVFIEINRNRNEG
      ISVRHNPEFT MMELMAYADYHDLIELTESLFRTLAQEVLGTTKVITYGEHVDFGKPFKELTMREAIKKYRPETDMADLDN
      FDAAKALAESIGITVE KSWGLGRIVTEIFDEVAEHLIQPTFITTEYPAEVSPLARRNDVNPEITDRFEFFIGGREIGNGF
      SELNDAEDQAERFQEQVNAKAAG DDEAMFYDEDYVTALEYGLPPTAGLGIGIDRMIMLFTNSHTIRDVILFPAMRP
      >1B8AA
      MYRTHYSSEITEELNGQKVKVAGWVWEVKDLGGIKFLWIRDRDGIVQITAPKKKVDPELFLKLI PKLRSEDVVA
      VEGVNVFT PKAKLGFELPEKIVVLNRAETPLPLDPTGKVKAEIDTRLNRRFMDLRRPEVMAIFKIRSSVFKAVR
      DFFHENGFIETPKI IATA TEGGTLEFPMKYFEEDAFLAESPOLYKEIMMASGLDRVYEIAPIFRAEEHNTTRHL
      NNEWSIDSEMAFIEDDEEVMSEFLERLVAHAI NYVREHNAKELDILNFELEPKLPFPRVSYDKALEILGDLGKEI
      PWGEDIDTEGERLLGKYMENENAPLYFLYQYPSEAKPFYIMK YDNKPEICRAFDEYRGVEISSGGQREHRH
      DILVEQIKEKGLNPESFEFYLKAFRYGMPHGGFGLGAERLIKQMLDLPNIREVILF PRDRRRLTP
    .....</aistls:requiresSequence>
    <aistls:hasOptions>-GAPOPEN=10 -GAPEXT=2</aistls:hasOptions>
  </aistls:ClustalWInput>
</rdf:RDF>
```

### ClustalW 入力 RDF

#### 1.3.2. 実行コマンド

---

利用するマシンから以下のコマンドを実行する。

% curl --data-binary @入力 RDF 名

http://semantic.medals.jp:8090/sadi-services/ClustalW -o 出力 RDF 名

### 1.3.3. 実行結果

---

以下の定義に従い、ClustalW の実行結果が RDF 形式で出力される。

- RDF のヘッダは以下となる。

```
<rdf:RDF
```

```
  xmlns:rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls "http://www.molprof.jp/ontologies/aistlssio.owl#"
```

- 主語は ClustalWOutput クラス、rdf:about 属性は入力 RDF ファイルで指定したものがそのまま指定される。

```
<aistls:ClustalWOutput
```

```
  rdf:about="http://www.molprof.jp/ontologies/clustalw.rdf#1">
```

- ClustalW のテキスト実行結果のトリプルは以下となる。

主語 : ClustalWOutput

述語 : requiresResultInTextFormat

目的語 : 文字列(ClustalW の実行結果)

```
<aistls:requiresResultInTextFormat>ClustalW 実行結果
```

```
</aistls:requiresResultInTextFormat>
```

ClustalW の出力結果の例を以下に示す。

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#">
  <aistls:ClustalWOutput rdf:about="http://www.molprof.jp/ontologies/clustalw.rdf#1">
    <aistls:requiresResultInTextFormat>CLUSTAL W (1.83) Multiple Sequence Alignments
    .....
    1ATIA          -WTPPRYFNMMFQDLRGPGRGRLAYLRPETAQGIFVNFKNVLDATSRKLGFGIAQIGK
    E64328        ELGEVKKFNLMFVTSIGPGGKR--TGYMRPETAQGFIFQFRRLAQFFRNKLPFGVVQIGK
    B64744        KFRDEVPRFRFGVMRSREFLMKDAYSFHTSQESLQETYDAMYAAYSKIIFSRMGLDFRAVQA
    E64454        -----HGKGTQLDVKLALRPTSETPIIYMMK-LWVKVHTDLPIKIYQIVN
    G64424        -----DHGGREMLRPEMTSPVVRFYLNELKNLQKPL--RLYYFAN
    1ADJA        -----DRGGRSLTLRPEGTAAMVRAYLEHGMKVWPQP-VRLWMAGP
    G64930        VDMCRGPHVPNMRFCCHHFKLMKTAGAYWRGDSNNKMLQRIYGTAWADKKALNAYLQRLEE
    D64449        KGHPLSELSRKIVAKEEKKEEGEESKFYLLNPETEEIIELNENNINIIKDEELLALAKHE
    1SESA        -----KALGEEAKRLEEALREKEARLEALLQVPLPPWPAGAVGGE
    A26400        -----KLGEELDAKAELDALQAEIRDIALTI PNLPADVEVPVQKD
    1PYSA        -----EGFRLEGPLGEEVEGRLLLRTHTSMPQVRYMVAHTP
    y|Pyrococcus -----LKAIVGLRKEGWAEVSKTKEGLTLKLSEKGGKAEKRAIDIALEVL
    JT0942        -----RPEMAQRLKTRAKITSLVRRFMDDHGFLDIETPMLTKATPE
    1B8AA        -----FTPKAKLGFELPEKIVVLNRAETPLPLDPTGKVKAEKD
    1ASZB        -----IVKKVDEPIKSATVQNLEIHITKIYTISETPEALPILLEASRSEAE
    1LYLA        -----DSLPEGVYNDQFKKWDLGDIIIGARGTLFKTQTGELS IHCTELRL
    S56383        -----FETRFVGPFGHSQGMNLWLMTSPEYHMKRLLVAGCGPVFQ
    .....
    .....
    (
    D64449:0.41624,
    G64930:0.40428)
    :0.01577)
    :0.01909);</aistls:requiresResultInTextFormat>
  </aistls:ClustalWOutput>
</rdf:RDF>
```

## ClustalW 出力結果

## 1.4. IPknot

---

### 1.4.1. 入力 RDF の準備

---

以下の方法で IPknot の入力 RDF を作成する。

- RDF のヘッダに使用語彙 RDF、AISTLS SIO OWL を以下のように定義する。

```
<rdf:RDF
```

```
  xmlns:rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls "http://www.molprof.jp/ontologies/aistlssio.owl#"
```

- 主語は IPknotInput クラス、rdf:about 属性は任意とする。

```
<aistls:IPknotInput rdf:about="http://www.molprof.jp/ontologies/ipknot.rdf#1">
```

- RNA 配列を入力とする場合、RNA 配列のトリプルを定義する。

主語 : IPknotInput

述語 : requiresQueryRNASequence

目的語 : 文字列(RNA 配列)

```
<aistls:requiresQueryRNASequence>RNA 配列
```

```
</aistls:requiresQueryRNASequence>
```

- ClustalW 形式のマルチプルアラインメントを入力とする場合、ClustalW 形式マルチプルアラインメントのトリプルを定義する。

主語 : IPknotInput

述語 : requiresClustalWMultipleAlignment

目的語 : 文字列(ClustalW 形式マルチプルアラインメント)

```
<aistls:requiresClustalWMultipleAlignment>ClustalW 形式マルチプルアライン
```

```
メント</aistls:requiresClustalWMultipleAlignment>
```

- コマンドラインオプションのトリプルを定義する。

主語 : IPknotInput

述語 : hasOptions

目的語 : 文字列(コマンドラインオプション)

```
<aistls:hasOptions>-i </aistls:hasOptions>
```

この定義に基づいて IPknot の入力 RDF を作成する。

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#">
  <aistls:IPknotInput rdf:about="http://www.molprof.jp/ontologies/ipknot.rdf#1">
    <aistls:requiresQueryRNASequence>>Tomato_mosaic_virus.1
      GUGUCUUGGAGCGCGCGGAGUAAACAUAUAUGGUUCAUAUAUGUCCGUAGGCACGUAAAAAAGCGA
    >Tobacco_mosaic_virus.1
      GUGUCUUGGAUCGCGCGGGUCAAAUGUAUAUGGUUCAUAUACAUCGCGAGGCACGUAAUAAA-GCGA
    >Rehmannia_mosaic_vir.1
      GUGUCUUGGUUCGCGCGGGUCAAGUGUAUAUGGUGCAUAUACAUCGCGUAGGCACGUAAUAAA-GCGA
    >B.pepper.1
      GUGUCUUGGAACGCGCGGGUCAAAUAUAAGUGGUUCACUUAUAUCGCGUAGGCACGAAAAAUU-GCGU</aistls:req
quiresQueryRNASequence>
    <aistls:hasOptions>-i</aistls:hasOptions>
  </aistls:IPknotInput>
</rdf:RDF>

```

## IPknot 用入力 RDF

### 1.4.2. 実行コマンド

---

利用するマシンから以下のコマンドを実行する。

```
% curl --data-binary @入力 RDF 名
```

```
http://semantic.medals.jp:8090/sadi-services/IPknot -o 出力 RDF 名
```

### 1.4.3. 実行結果

以下の定義に従い、IPknot の実行結果が RDF 形式で出力される。

- RDF のヘッダは以下となる。

```
<rdf:RDF
```

```
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#"
```

- 主語は IPknotOutput クラス、rdf:about 属性は入力 RDF ファイルで指定したものがそのまま指定される。

```
<aistls:IPknotOutput
```

```
  rdf:about="http://www.molprof.jp/ontologies/ipknot.rdf#1">
```

- IPknot の実行結果のトリプルは以下となる。

主語 : IPknotOutput

述語 : requiresResultInTextFormat

目的語 : 文字列(IPknot の実行結果)

```
<aistls:requiresResultInTextFormat>IPknot 実行結果
```

```
</aistls:requiresResultInTextFormat>
```

IPknot の出力結果の例を以下に示す。

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#"
  <aistls:IPknotOutput rdf:about="http://www.molprof.jp/ontologies/ipknot.rdf#1">
    <aistls:requiresResultInTextFormat>&gt;Tomato_mosaic_virus.1
    GUGUCUUGGAGCGCGCGGAGUAAACAUAUAUGGUUCAUAUAUGUCCGUAGGCACGUAAAAAAGCGA
    ((((((.....(((.....((((((((.....)))))))))))))))))).....
    &gt;Rehmannia_mosaic_vir.1
    GUGUCUUGGUUCGCGCGGUCAAGUGUAUAUGGUGCAUAUACAUCCGUAGGCACGUAAUAAA-GCGA
    (((((.....[[]])...((((((((.....)))))))).]]].))))).....
    .....</aistls:requiresResultInTextFormat>
  </aistls:IPknotOutput>
</rdf:RDF>
```

#### IPknot 出力結果

## 1.5. Mafft

---

### 1.5.1. 入力 RDF の準備

---

以下の方法で Mafft の入力 RDF を作成する。

- RDF のヘッダに使用語彙 RDF、AISTLS SIO OWL を以下のように定義する。

```
<rdf:RDF
```

```
  xmlns:rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls "http://www.molprof.jp/ontologies/aistlssio.owl#"
```

- 主語は MafftInput クラス、rdf:about 属性は任意とする。

```
<aistls:MafftInput rdf:about="http://www.molprof.jp/ontologies/mafft.rdf#1">
```

- 配列のトリプルを定義する。

```
  主語 : MafftInput
```

```
  述語 : requiresSequence
```

```
  目的語 : 文字列(核酸またはアミノ酸配列)
```

```
<aistls:requiresSequence>配列(3 つ以上の配列が必
```

```
要)</aistls:requiresSequence>
```

- コマンドラインオプションのトリプルを定義する。

```
  主語 : MafftInput
```

```
  述語 : hasOptions
```

```
  目的語 : 文字列(コマンドラインオプション)
```

```
<aistls:hasOptions>--retree 2 --maxiterate 0 --bl 62 --op 1.53 --ep 0.0
```

```
--clustalout</aistls:hasOptions>
```

この定義に基づいて Mafft の入力 RDF を作成する。



```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#">
  <aistls:MafftInput rdf:about="http://www.molprof.jp/ontologies/mafft.rdf#1">
    <aistls:requiresSequence>>1LYLA
      FNDELNRNREKLAALRQQGVAFPNDFRDHTSDQLHEEFDAKDNQELESLNIEVSVAGRM
      MTRRIMGKASFVTLQDVGGRIQLYVARDLPEGVYNDQFKKWDLDIIGARGTLFKTQTG
      ELSIHCTELRLLTKALRPLPDQEVRYRQRYLDLIANDKSRQTFVVRSKILAAIRQFMVAR
      EQVNAKAAGDDEAMFYDEDYVTALEYGLPPTAGLGIGIDRMIMLFTNSHTIRDVILFPAM
      RP
    >1B8AA
      MYRTHYSSEITEELNGQVKVAGVWVEVKDLGGIKFLWIRDRDQIVQITAPKKKVDPELF
      KLI PKLRSEDDVAVEGVNFTPKAKLGFELPEKIVVLNRAETPLPLDPTGKVKAEKDTR
      LNNRFMDLRRPEVMAIFKIRSSVFKAVRDFFHENGFIETPKIIATATEGGTELFPMKY
      PNIREVILFPRDRRRLTP
    >1ASZB
      EDTAKDNYGKPLIQSRSDRTGQKRVKFDLDEAKDSKEVLFRRVHNTRQQGATLAF
      LTLRQQASLIQGLVKANKEGTISKNMVKWAGSLNLESIVLVRGIVKKVDEPIKSATVQNL
      EIHITKIYTISETPEALPILLEASRSEAEAEAAGLPVVNLDTRLDIRVIDLRTVTNQAI
      FRIQAGVCELFREYLATKKFTEVHTPKLLGAPSEGGSSVFEVTFK GKAYLAQSPQFNKQ
      QLIVADFERVYEIGPVFRAENSNTHRMTEFTGLDMEMAFEEHYHEVLDTLSELFVFI FS
      KFLGKLVDRDKYDTDFYILDKFPLEIRPFYTMPDPANPKYSNSYDFFMRGEEILSGAQRIH
      .....
      .....
      .....
      EIYEKLGKFRVHIDDRDIRPGRKFNDWEIKGVPLRIEVGPKDIENTKKITLFRDRDTMEKF
      QVDETQLMEVVEKTLNNIMENIKNRAWKFEFENFITLEDINPDEIKNILSEKRGVILVPF
      KEEIYNEELEEKVEATILGETEYKGNKYIAIAKTY
    </aistls:requiresSequence>
    <aistls:hasOptions></aistls:hasOptions>
  </aistls:MafftInput>
</rdf:RDF>

```

## Mafft 用入力 RDF

## 1.5.2. 実行コマンド

---

利用するマシンから以下のコマンドを実行する。

```
% curl --data-binary @入力 RDF 名  
http://semantic.medals.jp:8090/sadi-services/Mafft -o 出力 RDF 名
```

## 1.5.3. 実行結果

---

以下の定義に従い、Mafft の実行結果が RDF 形式で出力される。

- RDF のヘッダは以下となる。  
`<rdf:RDF`  
    `xmlns:rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#"`  
    `xmlns:aistls "http://www.molprof.jp/ontologies/aistlssio.owl#"`
- 主語は MafftOutput クラス、`rdf:about` 属性は入力 RDF ファイルで指定したものがそのまま指定される。  
`<aistls:MafftOutput rdf:about="http://www.molprof.jp/ontologies/mafft.rdf#1">`
- Mafft の実行結果のトリプルは以下となる。  
主語 : MafftOutput  
述語 : `requiresResultInTextFormat`  
目的語 : 文字列(Mafft の実行結果)  
`<aistls:requiresResultInTextFormat>Mafft 実行結果`  
`</aistls:requiresResultInTextFormat>`

Mafft の出力結果の例を以下に示す。

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#">
  <aistls:MafftOutput rdf:about="http://www.molprof.jp/ontologies/mafft.rdf#1">
    <aistls:requiresResultInTextFormat>CLUSTAL format alignment by MAFFT FFT-NS-2
      (v6.717b)
      1LYLA      FNDELNRNRREKLAALRQQGVAFPNDFRRDHTSDQLHEEFDAKDNQLESLNIEVSVAGRM
      1B8AA      -----MYRTHY-----SSEITEELNGQKVKVAGWV
      1ASZB      -----EDTAKDNYGKLP LIQSRSDRTGQKRVKFVD-L
      1ADJA      -----
      1PYSA      -----
      1LYLA      -----LVVGGFER-----VFEINR-NFRNE-----
      1B8AA      -----MMASGLDR-----VYEIAP-IFRAE-----
      1ASZB      -----LIVADFER-----VYEIGP-VFRAE-----
      1ADJA      -----YLEHGMKVWPQ-----PVRLWMAGP-MFRAE-----
      1PYSA      -----MVAHTP-----PFRIVVPGR-VFRFE-----
                                                    *
      .....
      .....
      1LYLA      -----AMRP-----
      1B8AA      -RDRRLTP-----
      1ASZB      -RDPKRLRP-----
      1ADJA      FLGEDEL RAGEVTLKRLATGEQVRLSREEVPGYLLQALG
      1PYSA      ----KFLEQFKGVL-----
    </aistls:requiresResultInTextFormat>
  </aistls:MafftOutput>
</rdf:RDF>
```

**Mafft** 出力結果

## 1.6. Psipred

---

### 1.6.1. 入力 RDF の準備

---

以下の方法で Psipred の入力 RDF を作成する。

- RDF のヘッダに使用語彙 RDF、AISTLS SIO OWL を以下のように定義する。

```
<rdf:RDF
```

```
  xmlns:rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls "http://www.molprof.jp/ontologies/aistlssio.owl#"
```

- 主語は PsiPredInput クラス、rdf:about 属性は任意とする。

```
<aistls:PsiPredInput
```

```
  rdf:about="http://www.molprof.jp/ontologies/psipred.rdf#1">
```

- アミノ酸配列のトリプルを定義する。

```
  主語 : PsiPredInput
```

```
  述語 : requiresQueryProteinSequence
```

```
  目的語 : 文字列(アミノ酸配列)
```

```
<aistls:requiresQueryProteinSequence>
```

```
  アミノ酸配列
```

```
</aistls:requiresQueryProteinSequence>
```

この定義に基づいて Psipred の入力 RDF を作成する。

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#">
  <aistls:PsiPredInput rdf:about="http://www.molprof.jp/ontologies/psipred.rdf#1">
    <aistls:requiresQueryProteinSequence>>test
      MANLGCWMLVLFVATWSDLGLCKKRPKPGGWNTGGSRYPGQSPGGNRYPPQGGGGWGQPHGGGWQPHGGGWQPHGGGW
GQPHGGGWQGGGTHSQWNKPSKPKTNMKHMAGAAAAGAVVGLGGYMLGSAMSRPIIHFGSDYEDRYRENMHRYPNQVYYRPMDE
YSNQNNFVHDCVNITIKQHTVTTTTKGENFTETDVKMMERVVEQMCITQYERESQAYYQRGSSMVLFSPPVILLISFLIFLIVGMA
NLGCWMLVLFVATWSDLGLCKKRPKPGGWNTGGSRYPGQSPGGNRYPPQGGGGWGQPHGGGWQPHGGGWQPHGGGWQPHGGGW
GQGGGTHSQWNKPSKPKTNMKHMAGAAAAGAVVGLGGYMLGSAMSRPIIHFGSDYEDRYRENMHRYPNQVYYRPMDEYSNQNNFV
HDCVNIMAGAAAAGAVVGLGGYMLGSAMSRPIIHFGSDYEDRYRENMHRYPNQVYYRPMDEYSNQNNFVHDCVNITIKQHTVTTTT
TKGENFTETDVKMMERVVEQMCITQYERESQAYYQRGSSMVLFSPPVILLISFLIFLIVG</aistls:requiresQueryProt
einSequence>
    </aistls:PsiPredInput>
  </rdf:RDF>

```

## Psipred 用入力 RDF

### 1.6.2. 実行コマンド

---

利用するマシンから以下のコマンドを実行する。

% curl --data-binary @入力 RDF 名

http://semantic.medals.jp:8090/sadi-services/Psipred -o 出力 RDF 名

### 1.6.3. 実行結果

---

以下の定義に従い、Psipred の実行結果が RDF 形式で出力される。

- RDF のヘッダは以下となる。

```
<rdf:RDF
```

```
  xmlns:rdf “http://www.w3.org/1999/02/22-rdf-syntax-ns#”
```

```
  xmlns:aistls “http://www.molprof.jp/ontologies/aistlssio.owl#”>
```

- 主語は PsiPredOutput クラス、rdf:about 属性は入力 RDF ファイルで指定したものをそのまま指定する。

```
<aistls:PsiPredOutput
```

```
  rdf:about=”http://www.molprof.jp/ontologies/psipred.rdf#1”>
```

- Psipred の実行結果のトリプルは以下となる。

```
  主語 : PsiPredOutput
```

```
  述語 : requiresResultInTextFormat
```

```
  目的語 : 文字列(Psipred の実行結果)
```

```
<aistls:requiresResultInTextFormat>Psipred 実行結果
```

```
</aistls:requiresResultInTextFormat>
```

Psipred の出力結果の例を以下に示す。

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#">
  <aistls:PsiPredOutput rdf:about="http://www.molprof.jp/ontologies/psipred.rdf#1">
    <aistls:requiresResultInTextFormat># PSIPRED HFORMAT (PSIPRED V2.5 by David Jones)
    Conf: 976304410010420000111113889987678876789888999888879888999988
    Pred: CCCCCHHHHHHHHHHHHCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    AA: MANLGCWMLVLFVATWSDLGLCKKRPKPGGWNTGGSRYPGQSPGGNRYPPQGGGGWQP
          10      20      30      40      50      60

    Conf: 88899898889989888899898888767788735434688777333143200001124
    Pred: CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCHHHHCCCHHHHH
    AA: HGGGWGQPHGGGWGQPHGGGWGQPHGGGWGQGGGTHSQWNKPSKPKTNMKHMAGAAAAGA
          70      80      90      100     110     120

    Conf: 541000354322320732104774302456665564763125411233285454021100
    Pred: HHHCHHHHHHHHHCCCECCCCCHHHHHHHHHHHCCCCCECCCHHHCCCCCEEEEE
    AA: VVGGLGGYMLGSAMSRPIIHFGSDYEDRYRENMHRYPNQVYYRPMDEYSNQNNFVHDCV
          130     140     150     160     170     180

    .....
    .....

    Conf: 389999998764234432234654228605885697303222567721109
    Pred: HHHHHHHHHHHHHHHHHHHHHHHHHHHHHCCCEEEEECCCCCHHHHHHHHHHHCC
    AA: DVKMMERVVEQMCITQYERESQAYYQRGSSMVLFSPPVILLISFLIFLIVG
          970     980     990     1000    1010

    </aistls:requiresResultInTextFormat>
  </aistls:PsiPredOutput>
</rdf:RDF>
```

### Psipred 出力結果

## 1.7. Raccess

---

### 1.7.1. 入力 RDF の準備

---

以下の方法で Raccess の入力 RDF を作成する。

- RDF のヘッダに使用語彙 RDF、AISTLS SIO OWL を以下のように定義する。

```
<rdf:RDF
```

```
  xmlns:rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls "http://www.molprof.jp/ontologies/aistlssio.owl#"
```

- 主語は RaccessInput クラス、rdf:about 属性は任意とする。

```
<aistls:RaccessInput
```

```
  rdf:about="http://www.molprof.jp/ontologies/raccess.rdf#1">
```

- RNA 配列のトリプルを定義する。

主語 : RaccessInput

述語 : requiresQueryRNASequence

目的語 : 文字列(RNA 配列)

```
<aistls:requiresQueryRNASequence>RNA 配列
```

```
</aistls:requiresQueryRNASequence>
```

- コマンドラインオプションのトリプルを定義する。

主語 : RaccessInput

述語 : hasOptions

目的語 : 文字列(コマンドラインオプション)

```
<aistls:hasOptions>-access_len=50 </aistls:hasOptions>
```

この定義に基づいて Raccess の入力 RDF を作成する。



```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#"
  <aistls:RaccessInput rdf:about="http://www.molprof.jp/ontologies/raccess.rdf#1">
    <aistls:requiresQueryRNASequence>>gi|187607315|ref|NM_014909.4| Homo sapiens
vasohibin 1 (VASH1), mRNA
      GCCCTGCGCGCCGCCGAGCCGGTCCCGCTGAGCCGCGGGCCCCGTGCCCTGCGATGGCTCGGCTGGTG
      CAGCGCGGCGCCAGGTGCCAGCCGTCTCCCGCTGAGACGCGCCCGAGTGGGGACCCGCTGGGCTCGGG
      GCTCGCAGCCTTCGCCTCCCCGCGCGCCCGCTCCCTTCTGGGGACTCCGCCGCTGTTTCTGGGGACGA
      GGGGACAGGGGACCCAGACAAAGCCCACTTTGTGCAGGGAGTTGGCCGCAGGCGGGGAATGTGCGCGTCG
      GCGCGCGCCCCCTCCCCGCTCCCGGCCAGCTGGGAGTCTTGGCTCCCGGACTTGTCTCGTCGCGTCGGAG
      AAATCGCCCCCAGCGCCGCTCTCCCGCCGGGGTCTTGGTTCCGAGCTCGCGCGCCGGGAGTCGCCT
      CGGTCTTCTTGGGGCGCGCGCAGATGTGAGCGTGCAGAGTGTGTAGGGGATTTTGTTCCTCCGAAA
      CTGAGACCCAGGGCGCCAGTGGGCACCCGTGCCTTGACTCTGTCTTCTGCAGCCGCTGGTCCGAGCT
      GTCTGGCCTCAGTTTCCCTCCGACTTTTCTCCGCTCTGCCAGCCCTCACTGCTGCCCGTCATTGTTCTCG
      CAGTTAGATGGGGGTGCTTTGTGACGGCTGCCAAGTTGGGGTGTGTTCTTTATTCCGTTTTTCAAACA
      GAACAAGGCCTCCAAGGCTGACCCAGACAACCCACCCCTCGGACCCTAATTCACCTTATTGCACTGAT
      TTTTTTATCAAGTCGTATTTTATTGTACAGGACCACGCCCTGATTTCTTAAAGGCGCCTTGCACTCTG
      GCCATGTGTTATCTCTGCAGCCGGTGTGTGGGAGGCCCTTGTGAGCCAGTTGTTTTCCCGCCTCCACCA
      CCCCCCTCGAAGATTTAGGGATGCCAGGGGGAAGAAGGTGGCTGGGGGTGGCAGCAGCGGTGCCACTCC
      AACGTCGGCTGCGGCCACCGCCCCCTCTGGGGTCAGGCGTTTGGAGACCAGCGAAGGAACCTCAGCCCAG
      CCAAGCTGCTCTCGCTCCCACTGAGCCAAGCCCCCTAACTTTGGGCTAGAGGCGGTTAGTAT.....
    </aistls:requiresQueryRNASequence>
    <aistls:hasOptions></aistls:hasOptions>
  </aistls:RaccessInput>
</rdf:RDF>

```

## Raccess 用入力 RDF

### 1.7.2. 実行コマンド

利用するマシンから以下のコマンドを実行する。

```
% curl --data-binary @入力 RDF 名
```

```
http://semantic.medals.jp:8090/sadi-services/Raccess -o 出力 RDF 名
```

### 1.7.3. 実行結果

---

以下の定義に従い、Raccess の実行結果が RDF 形式で出力される。

- RDF のヘッダは以下となる。

```
<rdf:RDF
```

```
  xmlns:rdf “http://www.w3.org/1999/02/22-rdf-syntax-ns#”
```

```
  xmlns:aistls “http://www.molprof.jp/ontologies/aistlssio.owl#”>
```

- 主語は RaccessOutput クラス、rdf:about 属性は入力 RDF ファイルで指定したものがそのまま指定される。

```
<aistls:RaccessOutput
```

```
  rdf:about=”http://www.molprof.jp/ontologies/raccess.rdf#1”>
```

- Raccess の実行結果のトリプルは以下となる。

主語 : RaccessOutput

述語 : requiresResultInTextFormat

目的語 : 文字列(Raccess の実行結果)

```
<aistls:requiresResultInTextFormat>Raccess 実行結果
```

```
</aistls:requiresResultInTextFormat>
```

Raccess の出力結果の例を以下に示す。

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#"
  <aistls:RaccessOutput rdf:about="http://www.molprof.jp/ontologies/raccess.rdf#1">
    <aistls:requiresResultInTextFormat>&gt;gi|187607315|ref|NM_014909.4| Homo sapiens
vasohibin 1 (VASH1), mRNA
      6020   6029   0.62373
      6019   6028   0.623859
      6018   6027   0.623156
      6017   6026   0.622921
      6016   6025   0.648241
      6015   6024   1.34601
      6014   6023   2.08364
      6001   6010   9.45949
      6000   6009   8.523
      5999   6008   9.24843
      .....
      .....
      11     20     11.5702
      10     19     11.5887
      9      18     11.4777
      8      17     11.4848
      7      16     13.0533
      6      15     12.7788
      5      14     10.5224
      4      13     11.3123
      3      12     10.8754
      2      11     10.6341
      1      10     9.07663
      0      9      6.90453/aistls:requiresResultInTextFormat>
  </aistls:RaccessOutput>
</rdf:RDF>
```

**Raccess 出力結果**

## 1.8. RactIP

---

### 1.8.1. 入力 RDF の準備

---

以下の方法で RactIP の入力 RDF を作成する。

- RDF のヘッダに使用語彙 RDF、AISTLS SIO OWL を以下のように定義する。  
<rdf:RDF  
    xmlns:rdf “http://www.w3.org/1999/02/22-rdf-syntax-ns#”  
    xmlns:aistls “http://www.molprof.jp/ontologies/aistlssio.owl#”>
- 主語は RactIPInput クラス、rdf:about 属性は任意とする。  
<aistls:RactIPInput rdf:about=”http://www.molprof.jp/ontologies/ractip.rdf#1”>
- 問い合わせ RNA 配列のトリプルを定義する。  
主語 : RactIPInput  
述語 : requiresQueryRNASequence  
目的語 : 文字列(RNA 配列)  
<aistls:requiresQueryRNASequence>RNA 配列  
</aistls:requiresQueryRNASequence>
- ターゲット RNA 配列のトリプルを定義する。  
主語 : RactIPInput  
述語 : requiresSubjectRNASequence  
目的語 : 文字列(RNA 配列)  
<aistls:requiresSubjectRNASequence>RNA 配列  
</aistls:requiresSubjectRNASequence>
- コマンドラインオプションのトリプルを定義する。  
主語 : RactIPInput  
述語 : hasOptions  
目的語 : 文字列(コマンドラインオプション)  
<aistls:hasOptions>-i</aistls:hasOptions>

この定義に基づいて RactIP の入力 RDF を作成する。

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#">
  <aistls:RactIPInput rdf:about="http://www.molprof.jp/ontologies/ractip.rdf#1">
    <aistls:requiresQueryRNASequence>>R1inv
    GGCAACGGAUGGUUCGUUGCC</aistls:requiresQueryRNASequence>
    <aistls:requiresSubjectRNASequence>>R2inv
    GCACCGAACCAUCCGGUGC</aistls:requiresSubjectRNASequence>
    <aistls:hasOptions>-i</aistls:hasOptions>
  </aistls:RactIPInput>
</rdf:RDF>
```

## RactIP 用入力 RDF

### 1.8.2. 実行コマンド

---

利用するマシンから以下のコマンドを実行する。

% curl --data-binary @入力 RDF 名

http://semantic.medals.jp:8090/sadi-services/RactIP -o 出力 RDF 名

### 1.8.3. 実行結果

---

以下の定義に従い、RactIP の実行結果が RDF 形式で出力される。

- RDF のヘッダは以下となる。

```
<rdf:RDF
```

```
  xmlns:rdf “http://www.w3.org/1999/02/22-rdf-syntax-ns#”
```

```
  xmlns:aistls “http://www.molprof.jp/ontologies/aistlssio.owl#”>
```

- 主語は RactIPOutput クラス、rdf:about 属性は入力 RDF ファイルで指定したものがそのまま指定される。

```
<aistls:RactIPOutput
```

```
  rdf:about=”http://www.molprof.jp/ontologies/ractip.rdf#1”>
```

- RactIP の実行結果のトリプルは以下となる。

主語 : RactIPOutput

述語 : requiresResultInTextFormat

目的語 : 文字列(RactIP の実行結果)

```
<aistls:requiresResultInTextFormat>RactIP 実行結果
```

```
</aistls:requiresResultInTextFormat>
```

RactIP の出力結果の例を以下に示す。

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#"
  <aistls:RactIPOutput rdf:about="http://www.molprof.jp/ontologies/ractip.rdf#1">
    <aistls:requiresResultInTextFormat>*    0:  objval =  0.000000000e+00  infeas =
0.000000000e+00 (0)
    *   36:  objval =  1.604034567e+01  infeas =  0.000000000e+00 (0)
    OPTIMAL SOLUTION FOUND
    Integer optimization begins...
    +   36: mip =    not found yet &lt;=          +inf          (1; 0)
    +   36: &gt;&gt;&gt;&gt;&gt;&gt;  1.604034567e+01 &lt;=  1.604034567e+01  0.0% (1; 0)
    +   36: mip =  1.604034567e+01 &lt;=    tree is empty  0.0% (0; 1)
    INTEGER OPTIMAL SOLUTION FOUND
    &gt;R1inv
    GGCAACGGAUGGUUCGUUGCC
    ((((((([[[[[[]]]]]))))))
    &gt;R2inv
    GCACCGAACCAUCCGUGC
    (((([[]]]]]]]))))</aistls:requiresResultInTextFormat>
  </aistls:RactIPOutput>
</rdf:RDF>
```

### RactIP 出力結果

## 1.9. Wolfpsort

---

### 1.9.1. 入力 RDF の準備

---

以下の方法で Wolfpsort の入力 RDF を作成する。

- RDF のヘッダに使用語彙 RDF、AISTLS SIO OWL を以下のように定義する。

```
<rdf:RDF
```

```
  xmlns:rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls "http://www.molprof.jp/ontologies/aistlssio.owl#"
```

- 主語は WolfPsortInput クラス、rdf:about 属性は任意とする。

```
<aistls:WolfPsortInput
```

```
  rdf:about="http://www.molprof.jp/ontologies/wolfpsort.rdf#1">
```

- 界情報のトリプルを以下のように定義する。

```
  主語 : WolfPsortInput
```

```
  述語 : requiresKingdomInformation
```

```
  目的語 : 文字列(animal、plant、fungi のうちどれか一つ)
```

```
<aistls:requireKingomdInfomationOf>文字列
```

```
</aistls:requireKingdomInformationOf>
```

- アミノ酸配列のトリプルを定義する。

```
  主語 : WolfPsortInput
```

```
  述語 : requiresQueryProteinSequence
```

```
  目的語 : 文字列(アミノ酸配列)
```

```
<aistls:requiresQueryProteinSequence>
```

```
  アミノ酸配列
```

```
</aistls:requiresQueryProteinSequence>
```

この定義に基づいて Wolfpsort の入力 RDF を作成する。



```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#">
  <aistls:WolfPsortInput rdf:about="http://www.molprof.jp/ontologies/wolfpsort.rdf#1">
    <aistls:requiresKingdomInformationOf>animal</aistls:requiresKingdomInformationOf>
    <aistls:requiresQueryProteinSequence>MANLGCWMLVLFVATWSDLGLCKKRPKPGGWNTGGSRYPGQGSF
GGNRYPPQGGGGWGQPHGGGWGQPHGGGWGQPHGGGWGQPHGGGWGQGGGTHSQWNKPSKPKTNMKHMAGAAAAGAVVGGGLGGYMLG
SAMSRPPIIHFGSDYEDRYRENMHRYPNQVYYRPMDEYSNQNNFVHDCVNITIKQHTVTTTTKGENFTETDVKMMERVVEQMCITQY
ERESQAYYQRGSSMVLFSPPVILLISFLIFLIVG</aistls:requiresQueryProteinSequence>
  </aistls:WolfPsortInput>
</rdf:RDF>
```

## Wolfpsort 用入力 RDF

### 1.9.2. 実行コマンド

---

利用するマシンから以下のコマンドを実行する。

% curl --data-binary @入力 RDF 名

http://semantic.medals.jp:8090/sadi-services/Wolfpsort -o 出力 RDF 名

### 1.9.3. 実行結果

---

以下の定義に従い、Wolfpsort の実行結果が RDF 形式で出力される。

- RDF のヘッダは以下とする。

```
<rdf:RDF
```

```
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#"
```

- 主語は WolfPsortOutput クラス、rdf:about 属性は入力 RDF ファイルで指定したものをそのまま指定する。

```
<aistls:WolfPsortOutput
```

```
  rdf:about="http://www.molprof.jp/ontologies/wolfpsort.rdf#1">
```

- Wolfpsort の実行結果のトリプルは以下となる。

主語 : WolfPsortOutput

述語 : requiresResultInTextFormat

目的語 : 文字列(Wolfpsort の実行結果)

```
<aistls:requiresResultInTextFormat>Wolfpsort 実行結果
```

```
</aistls:requiresResultInTextFormat>
```

Wolfpsort の出力結果の例を以下に示す。

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#"
  <aistls:WolfPsortOutput rdf:about="http://www.molprof.jp/ontologies/wolfpsort.rdf#1">
    <aistls:requiresResultInTextFormat># k used for kNN is: 32
    queryProtein extr 19, E.R. 4, pero 4, lyso 3, E.R._mito 3, mito_pero 3
    </aistls:requiresResultInTextFormat>
  </aistls:WolfPsortOutput>
</rdf:RDF>
```

#### Wolfpsort 出力結果

## 2. Async 非同期型解析サービスについて

---

### 2.0. Async 同期型解析サービスの利用方法

---

Async 同期型解析サービス (Last, Modelling, PoodleL, PoodleS, Rassie) の場合、以下の3つの手順で各解析サービスを実行する。

- 1) ポーリング URI の取得

```
% curl --data-binary @入力 RDF 名  
http://semantic.medals.jp:8090/sadi-services/解析サービス名 (表 1-A 参照)
```

例えば、input.rdf を入力 RDF として用意し、PoodleL を実行する場合、以下のコマンドとなる。

```
% curl --data-binary @input.rdf  
http://semantic.medals.jp:8090/sadi-services/PoodleL
```

ポーリング URI を格納した RDF が標準出力される。

```
<rdf:RDF  
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"  
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#">  
  <aistls:PoodleLOutput rdf:about="http://www.molprof.jp/ontologies/poodlel.rdf#1">  
    <rdfs:isDefinedBy  
      rdf:resource="http://semantic.medals.jp:8090/sadi-services/PoodleL?poll=1"/>  
    </aistls:PoodleLOutput>  
</rdf:RDF>
```

ポーリング URI(赤文字部分)を含む RDF

- 2) SADI サーバへポーリング

```
% curl http://semantic.medals.jp:8090/sadi-services/解析サービス名?poll=任意の 16  
進数
```

例えば 1) の PoodleL を実行した場合は以下のコマンドとなる。

```
%curl http://semantic.medals.jp:8090/sadi-services/PoodleL?poll=1
```

サービスの実行が終了すると、COMPLETE:出力結果 URL が標準出力される。

```
% curl http://semantic.medals.jp:8090/sadi-services/PoodleL?poll=1  
COMPLETE: http://semantic.medals.jp/tmp/xxx/yyyy/poodleLResult.rdf
```

### 3) 実行結果の取得

```
% curl http://semantic.medals.jp/tmp/xxx/yyyy/ poodleLResult.rdf -o 出力 RDF 名
```

入力 RDF フォーマット、出力 RDF フォーマットについては、章 1.0 の Sync 同期型解析サービスの利用方法を参考にしてください。

## 2.1. Last

---

### 2.1.1. 入力 RDF の準備

---

以下の方法で Last の入力 RDF を作成する。

- RDF のヘッダに使用語彙 RDF、AISTLS SIO OWL を以下のように定義する。

```
<rdf:RDF
```

```
  xmlns:rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls "http://www.molprof.jp/ontologies/aistlssio.owl#"
```

- 主語は LastInput クラス、rdf:about 属性は任意とする。

```
<aistls:LastInput rdf:about="http://www.molprof.jp/ontologies/last.rdf#1">
```

- 問い合わせ配列のトリプルを定義する。

```
  主語 : LastInput
```

```
  述語 : requiresQuerySequence
```

```
  目的語 : 文字列(問い合わせ配列)
```

```
<aistls:requiresQuerySequence>   問   い   合   っ   せ   配   列
```

```
</aistls:requiresQuerySequence>
```

- ターゲット配列のトリプルを定義する。

```
  主語 : LastInput
```

```
  述語 : requiresSubjectSequence
```

```
  目的語 : 文字列(ターゲット配列)
```

```
<aistls:requiresSubjectSequence>ターゲット配列
```

```
</aistls:requiresSubjectSequence>
```

- Lastdb 用コマンドラインオプションのトリプルを定義する。

```
  主語 : LastInput
```

```
  述語 : hasOptionsForLastdb
```

```
  目的語 : 文字列(Lastdb 用コマンドラインオプション)
```

```
<aistls:hasOptionsForLastdb>-m110 -w1</aistls:hasOptionsForLastdb>
```

- Lastal 用コマンドラインオプションのトリプルを定義する。

```
  主語 : LastInput
```

```
  述語 : hasOptionsForLastal
```

```
  目的語 : 文字列(Lastal 用コマンドラインオプション)
```

```
<aistls:hasOptionsForLastal>-j4 -u0 -m10 -l1 -k1 -w0 -g1.0 -s2 -e30
</aistls:hasOptionsForLastal>
```

この定義に基づいて Last の入力 RDF を作成する。

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#">
  <aistls:LastInput rdf:about="http://www.molprof.jp/ontologies/last.rdf#1">
    <aistls:requiresQuerySequence>>gi|14518450|ref|NC_000868.1| Pyrococcus abyssi GE5,
complete genome
  GGGCTTTAGCCTCCTTCACCGCTTCCACGATTTTCTGCCTGTCAAAGGCATTCTAGACATCCCTCCTTA
  GGTTTTAAATTAATAAATCAAGGTGGAGTAAAAAGGGATGTTTTTAAATTTTCTCACTCTTCTCGGCC
  TTCTCAAATAGCTCGTCGTAAACCCCTTCATCTATTTCTCTCTGAACTTCCCTTGGATCCTTGCCTTCGA
  CGGTAACCTCCCATGCTTAAAGCCGTTCCAATGACTTCCTTGGCGGCAGCCTTAAGAGTCAATGCTAGCAT
  CTGGTTTCTCTTCATCTTAGCTATCTTGATAACTTGCTCCATCGTTAAGTTCCCAACGATATTGTGCTTC
  CTATCTCGAACTGCTTGGTTACTGGATCTACGATGATCTTCACTGGGACCTGCATCCCAGCGAACTCTTT
  .....
  AACATCAAGAAGCTGACCTACCACGGCCCTGAACTTCCTAGGATCTCCATGTCCCTCATCCTCTTCTTCA</aistls:re
quiresQuerySequence>
  <aistls:requiresSubjectSequence>
  AGATCCTTAGCCTTGTTGTTCTTTTCTCAAGGAGCTTTACGCTACCGTCTTACAGATCTCATAGATCG
  CGAAAACTCTGAATCTCCGTAGTGAGCGTCTATGAGATGTTTCATCATCTCCATTCCAAAGGCTACCTT
  .....
  GTTCTGGGACCTAGGTATCTACCGAGGTACCTACCTATCTTGGGCATTAATGGGGCCTCAGCTATGAAG
  TAATAACGTCAAGCCGAGCCTCCTCGCCGCTTCGGCAACTGCACCATCAGCGATGACCGCATCTTTAC
  TCTTTAAGTTCACTGCCACCTCGACACTCTGTGTGAAGTTACGCGGCTTGGCCC</aistls:requiresSubjectSe
quence>
  <aistls:hasOptionsForLastdb>-m110 -w1</aistls:hasOptionsForLastdb>
  <aistls:hasOptionsForLastal>-j4 -u0 -m10 -l1 -k1 -w0 -g1.0 -s2
-e30</aistls:hasOptionsForLastal>
  </aistls:LastInput>
</rdf:RDF>
```

## Last 用入力 RDF

## 2.1.2. 実行コマンド

---

利用するマシンから以下のコマンドを実行する。

```
% curl --data-binary @入力 RDF 名
http://semantic.medals.jp:8090/sadi-services/Last
```

Last は非同期通信となるため、まずポーリングに必要な URI を格納した RDF が返却される。次に、この RDF の isDefinedBy で定義された”http://semantic.medals.jp:8090/sadi-services/Last?poll=任意の 16 進数”を用いて SADI サーバへのポーリングを行う。

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#"
  <aistls:LastOutput rdf:about="http://www.molprof.jp/ontologies/last.rdf#1">
    <rdfs:isDefinedBy
rdf:resource="http://semantic.medals.jp:8090/sadi-services/Last?poll=8"/>
    </aistls:LastOutput>
</rdf:RDF>
```

### ポーリングのための URI を格納した RDF

```
% curl http://semantic.medals.jp:8090/sadi-services/Last?poll=8 (上図の場合 8)
%
```

Last が終了すると、Last の実行結果を格納した URL が標準出力に返却される。

```
% curl http://semantic.medals.jp:8090/sadi-services/Last?poll=8 (上図の場合 8)
COMPLETE: http://semantic.medals.jp/tmp/xxx/yyyy/seq1_seq2Result.rdf
%
```

出力された URL を用いて、以下のコマンドで結果を取得する。

```
% curl http://semantic.medals.jp/tmp/xxx/yyyy/seq1_seq2Result.rdf -o 出力 RDF 名
```

### 2.1.3. 実行結果

---

以下の定義に従い、Last の実行結果が RDF 形式で出力される。

- RDF のヘッダは以下となる。

```
<rdf:RDF
```

```
  xmlns:rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls "http://www.molprof.jp/ontologies/aistlssio.owl#"
```

- 主語は LastOutput クラス、rdf:about 属性は入力 RDF ファイルで指定されたものがそのまま指定される。

```
<aistls:LastOutput rdf:about="http://www.molprof.jp/ontologies/last.rdf#1">
```

- Last の実行結果のトリプルは以下となる。

主語 : LastOutput

述語 : requiresResultInTextFormat

目的語 : 文字列(Last の実行結果)

```
<aistls:requiresResultInTextFormat>Last 実行結果
```

```
</aistls:requiresResultInTextFormat>
```

- Last の PNG 画像->Base64 変換結果のトリプルは以下となる。

主語 : LastOutput

述語 : requiresResultInBase64BinaryFormat

目的語 : 文字列(Last PNG 画像->Base64 変換結果)

```
<aistls:requiresResultInBase64BinaryFormat>Last PNG 画像->Base64 変換結果
```

```
</aistls:requiresResultInBase64BinaryFormat>
```

\*入力 RDF ファイルの `lastalOptions` に”noimage”を追加すると、このトリプルは出力されない。



Last の出力結果の例を以下に示す。

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#"
  <aistls:LastOutput rdf:about="http://www.molprof.jp/ontologies/last.rdf#1">
    <aistls:requiresResultInBase64BinaryFormat>iVBORw0KGgoAAAANSUhEUgAAA8cAAAPoCAAAAA
DWntNlAAANoEleQVR42u3dAXIjRRZF0dz/ZmsJ
    oHbLq1LZbURnNc730CcmGMIw0GP7kvm/RHlsQLrhUWA65k9+dcaH37/91/jh/SPjw8fHGL/6q42v
    /lBjH3/7ya8KHfP61+ie71tQj98b93xPH79/5Bf/UBjvf/J4/EUf/6uv//Mifvj00TGvHcRjPBdz
    .....
    .....
    AAAASUVORK5CYII=</aistls:requiresResultInBase64BinaryFormat>
    <aistls:requiresResultInTextFormat># LAST version 58
    #
    # a=7 b=1 c=100000 e=30 d=18 x=27 y=10
    # u=0 s=2 m=10 l=1 k=1 i=134217728 w=0 t=0.910239 g=1 j=4
    # seq1
    .....
    .....

    a score=30
    s gi|14518450|ref|NC_000868.1|                                29878 36 +
139440GTTGAGAGTTCCAATAAGACTAAAATAGGATTGAAA
    s GATCCTTAGCCTTGTTGTTCTTTTCTCAAGGAGCTTTACGCTACCGTCTTCACAGATCTCATAGATCG 130267 36 -
144325 GTGGCGAGTTCCAATAAGACTAAAATAGAATTGAAA
    p 0.551 0.673 0.714 0.836 0.878 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0.999
0.999 0.998 0.996 0.99 0.972 0.916 0.749

    # CPU time: 0.36 seconds
    </aistls:requiresResultInTextFormat>
  </aistls:LastOutput>
</rdf:RDF>
```

Last 出力結果

## 2.2. Modelling

---

### 2.2.1. 入力 RDF の準備

---

以下の方法で Modelling の入力 RDF を作成する。

- RDF のヘッダに使用語彙 RDF、AISTLS SIO OWL を以下のように定義する。

```
<rdf:RDF
```

```
  xmlns:rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls "http://www.molprof.jp/ontologies/aistlssio.owl#"
```

- 主語は ModellingInput クラス、rdf:about 属性は任意とする。

```
<aistls:ModellingInput
```

```
  rdf:about="http://www.molprof.jp/ontologies/modelling.rdf#1">
```

- アミノ酸配列のトリプルを定義する。

```
  主語 : ModellingInput
```

```
  述語 : requiresQueryProteinSequence
```

```
  目的語 : 文字列(アミノ酸配列)
```

```
<aistls:requiresQueryProteinSequence>アミノ酸配列
```

```
</aistls:requiresQueryProteinSequence>
```

- BLAST 名のトリプルを定義する。

```
  主語 : ModellingInput
```

```
  述語 : requiresBlastProgramName
```

```
  目的語 : 文字列(BLAST または PSI-BLAST)
```

```
<aistls:requiresBlastProgramName>PSI-BLAST</aistls:requiresBlastProgram
Name>
```

- Iteration 数のトリプルを定義する (PSI-BLAST 選択時のみ)。

```
  主語 : ModellingInput
```

```
  述語 : setupIterationNumber
```

```
  目的語 : 整数(Iteration 数(>1 を指定))
```

```
<aistls:setupIterationNumber>3</aistls:setupIterationNumber>
```

- E-Value のトリプルを定義する。

```
  主語 : ModellingInput
```

```
  述語 : setupEvalueThreshold
```

目的語：倍精度数(E-Value)

```
<aistls:setupEvalueThreshold>0.00001</aistls:setupEvalueThreshold>
```

- ヒット領域カバレッジ閾値のトリプルを定義する。

主語：ModellingInput

述語：setupHitRegionCoverageThreshold

目的語：倍精度数(ヒット領域カバレッジの閾値)

```
<aistls:setupHitRegionCoverageThreshold>60.0
```

```
</aistls:setupHitRegionCoverageThreshold>
```

- ヒット領域一致度閾値のトリプルを定義する。

主語：ModellingInput

述語：setupHitRegionIdentityThreshold

目的語：倍精度数(ヒット領域一致度の閾値)

```
<aistls:setupHitRegionIdentityThreshold>30.0
```

```
</aistls:setupHitRegionIdentityThreshold>
```

- 最小配列長閾値のトリプルを定義する。

主語：ModellingInput

述語：setupMinimumSequenceLength

目的語：整数(最小配列長の閾値)

```
<aistls:setupMinimumSequenceLength>30
```

```
</aistls:setupMinimumSequenceLength>
```

- テンプレートカバレッジ閾値のトリプルを定義する。

主語：ModellingInput

述語：setupTemplateCoverageThreshold

目的語：倍精度数(テンプレートカバレッジの閾値)

```
<aistls:setupTemplateCoverageThreshold>90.0
```

```
</aistls:setupTemplateCoverageThreshold>
```

- テンプレート一致度閾値のトリプルを定義する。

主語：ModellingInput

述語：setupTemplateIdentityThreshold

目的語：倍精度数(テンプレート一致度の閾値)

```
<aistls:setupTemplateIdentityThreshold>90.0
```

```
</aistls:setupTemplateIdentityThreshold>
```

- MODELLER ライセンスキーのトリプルを定義する。

主語：ModellingInput

述語：requiresLicenseKey

目的語：文字列(MODELLER ライセンスキー)

<aistls:requiresLicenseKey>\*\*\*</aistls:requiresLicenseKey>

\*MODELLER ライセンスキーがないと Modelling は利用できません。

MODELLER の Web サイト(<http://saliab.org/modeller/registration.html>)より、  
ライセンスキー取得のための登録を行ってください。

- 生成モデル数閾値のトリプルを定義する。

主語 : ModellingInput

述語 : setupModelNumber

目的語 : 整数(生成モデル数の閾値)

<aistls:setupModelNumber>10</aistls:setupModelNumber>

この定義に基づいて Modelling の入力 RDF を作成する。

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#">
  <aistls:ModellingInput rdf:about="http://www.molprof.jp/ontologies/modelling.rdf#1">
    <aistls:requiresQueryProteinSequence>>sample
      MNGTEGPNFYVFPFSNKTGVVRSPEAPQYYLAEPWQFSMLAAYMFLIMLGFPINFLTLVTVQHKKLRTPLNYILLNLAV
      ADLFMVFGGFTTTLTYSLHGYPVFGPTGCNLEGFFATLGGEIALWSLVVLAIERVYVVKPMSNFRFGENHAIMGVAFTWVMALACA
      APPLVGWSRYIPEGMQCSCGIDYYTPHEETNNESFVIYMFVVFHFIIPLIVIFFCYGQLVFTVKEAAAQQQESATTQKAEKEVTRMVI
      IMVIAFLICWLPYAGVAFYIFTHQGSDFGPIFMTIPAFFAKTSAVYNPVIYIMMNKQFRNCMVTTLCCGKNKREIRLMKNREAAREC
      RRKKKEYVKCLENRAVLENQNKTLEELKTLKDLYSNKMSEEGPQVKIREASKDNVDFILSNVDLAMANSRRVMIAEIPTLAIDS
      VEVETNTTVLADEFIAHRLGLIPLQSMIDIEQLEYSRDCFCEHDCKCSVVLTLQAFGESESTNVYSKDLVIVSNLMGRNIGHPIIQ
      DKEGNGVLICKLRKGQELKLTCAKKGIAKEHAKWGPAAAEFEYDPWNKHKHTDYWYEQDSAKEWPQSKNCEYEDPPNEGDPFDYK
      AQADTFYMNVESVGSIPVDQVVVRGIDTLQKKVASILLALTQMDQDKVNFASGDNNTASNMLGSNEDVMMTGAEQDPYSNASQMGNT
      GSGGYDNAW</aistls:requiresQueryProteinSequence>
    <aistls:requiresBlastProgramName>BLAST</aistls:requiresBlastProgramName>
    <aistls:setupEvaluateThreshold>0.00005</aistls:setupEvaluateThreshold>
    <aistls:setupHitRegionCoverageThreshold>60.0</aistls:setupHitRegionCoverageThresh
old>
    <aistls:setupHitRegionIdentityThreshold>30.0</aistls:setupHitRegionIdentityThresh
old>
    <aistls:setupMinimumSequenceLength>30</aistls:setupMinimumSequenceLength>
    <aistls:setupTemplateCoverageThreshold>95.0</aistls:setupTemplateCoverageThreshol
d>
    <aistls:setupTemplateIdentityThreshold>95.0</aistls:setupTemplateIdentityThreshol
d>
  </aistls:ModellingInput>
</rdf:RDF>

```

## Modelling 用入力 RDF

## 2.2.2. 実行コマンド

---

利用するマシンから以下のコマンドを実行する。

```
% curl --data-binary @入力 RDF 名
http://semantic.medals.jp:8090/sadi-services/Modelling
```

Modelling は非同期通信となるため、まずポーリングに必要な URI を格納した RDF が返却される。次に、この RDF の isDefinedBy で定義された”http://semantic.medals.jp:8090/sadi-services/Modelling?poll=任意の 16 進数”を用いて SADI サーバへのポーリングを行う。

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#"
  <aistls:ModellingOutput rdf:about="http://www.molprof.jp/ontologies/modelling.rdf#1">
    <rdfs:isDefinedBy
rdf:resource="http://semantic.medals.jp:8090/sadi-services/Modelling?poll=10"/>
    </aistls:ModellingOutput>
</rdf:RDF>
```

### ポーリングのための URI を格納した RDF

```
% curl http://semantic.medals.jp:8090/sadi-services/Modelling?poll=10 (上図の場合
10)
%
```

Modelling が終了すると、Modelling の実行結果を格納した URL が標準出力に返却される。

```
% curl http://semantic.medals.jp:8090/sadi-services/Modelling?poll=10
COMPLETE: http://semantic.medals.jp/tmp/xxx/yyyy/modellingResult.rdf
%
```

出力された URL を用いて、以下のコマンドで結果を取得する。

```
% curl http://semantic.medals.jp/tmp/xxx/yyyy/modellingResult.rdf -o 出力 RDF 名
```

### 2.2.3. 実行結果

---

以下の定義に従い、Modelling の実行結果が RDF 形式で出力される。

- RDF のヘッダは以下となる。

```
<rdf:RDF
```

```
  xmlns:rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls "http://www.molprof.jp/ontologies/aistlssio.owl#"
```

- 主語は ModellingOutput クラス、rdf:about 属性は入力 RDF ファイルで指定したものがそのまま指定される。

```
<aistls:ModellingOutput
```

```
  rdf:about="http://www.molprof.jp/ontologies/modelling.rdf#1">
```

- アミノ酸配列のトリプルは以下となる。

主語 : ModellingOutput

述語 : requiresQueryProteinSequence

目的語 : 文字列(アミノ酸配列)

```
<aistls:requiresQueryProteinSequence>アミノ酸配列
```

```
</aistls:requiresQueryProteinSequence>
```

- BLAST 実行結果のトリプルは以下となる。

主語 : ModellingOutput

述語 : requiresBlastResult

目的語 : 文字列(BLAST 実行結果)

```
<aistls:requiresBlastResult>BLAST 実行結果
```

```
</aistls:requiresBlastResult>
```

- ヒット情報抽出結果のトリプルは以下となる。

主語 : ModellingOutput

述語 : requiresHitRegionInformation

目的語 : 文字列(ヒット情報抽出結果)

```
<aistls:requiresHitRegionInformation>ヒット情報抽出結果
```

```
</aistls:requiresHitRegionInformation>
```

- MODELLER 実行結果のトリプルを定義する。

主語 : ModellingOutput

述語 : requiresModellerResult

目的語 : 文字列(MODELLER 実行結果)



<aistls:requiresModellerResult>MODELLER 実行結果

</aistls:requiresModellerResult>

Modelling の出力結果の例を以下に示す。

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#"
  <aistls:ModellingOutput rdf:about="http://www.molprof.jp/ontologies/modelling.rdf#1">
    <aistls:requiresBlastResult>
      PDB      QHitRegionLen  QCov  QHitRange      THitRegionLen  Coverage
THitRange  Identity      E-value
      1qm0A  104    41.11  125-228 104    100.00  1-104  100.00  3.20122e-60
      1xyuA  111    43.87  121-231 111    100.00  1-111  90.99   1.43738e-59
      .....</aistls:requiresBlastResult>
      <aistls:requiresHitRegionInformation>PDB      QHitRegionLen  QCov  QHitRange
THitRegionLen  TCov  THitRange      Identity      E-value
      1qm0A  104    41.11  125-228 104    100.00  1-104  100.00  3.20122e-60
    </aistls:requiresHitRegionInformation>
      <aistls:requiresModellerResult>MODEL:1      Query hit region:1-326
EXPDTA  THEORETICAL MODEL, MODELLER 9v5 2012/02/09 21:51:43
REMARK  6 MODELLER OBJECTIVE FUNCTIO
N:      1915.0485
REMARK  6 MODELLER BEST TEMPLATE % SEQ ID: 100.000
ATOM    252  ND2  ASN   355    57.353  10.420  32.011  1.00112.53      N
ATOM    253  C    ASN   355    55.628  8.704  34.015  1.00112.53      C
ATOM    254  O    ASN   355    54.440  8.414  33.908  1.00112.53      O
      .....
      .....</aistls:requiresModellerResult>
      <aistls:requiresQueryProteinSequence>&gt;sp|P04156|PRIO_HUMAN Major prion protein
OS=Homo sapiens GN=PRNP PE=1 SV=1
      MANLGCWMLVLFVATWSDLGLCKKRPKPGGWNTGGSRYPGQSPGGNRYPPQGGGGWGQP
      HGGGGWGQPHGGGGWQPHGGGGWQPHGGGGWQGGGTHSQWNKPSKPKTNMKHMAGAAAAGA</aistls:requiresQueryPr
oteinSequence>
    </aistls:ModellingOutput>
  </rdf:RDF>
```

Modelling 出力結果

## 2.3. PoodleL

---

### 2.3.1. 入力 RDF の準備

---

以下の方法で PoodleL の入力 RDF を作成する。

- RDF のヘッダに使用語彙 RDF、AISTLS SIO OWL を以下のように定義する。

```
<rdf:RDF
```

```
  xmlns:rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls "http://www.molprof.jp/ontologies/aistlssio.owl#"
```

- 主語は PoodleLInput クラス、rdf:about 属性は任意とする。

```
<aistls:PoodleLInput
```

```
  rdf:about="http://www.molprof.jp/ontologies/poodlel.rdf#1">
```

- アミノ酸配列のトリプルを定義する。

```
  主語 : PoodleLInput
```

```
  述語 : requiresQueryProteinSequence
```

```
  目的語 : 文字列(アミノ酸配列)
```

```
<aistls:requiresQueryProteinSequence>
```

```
  アミノ酸配列
```

```
</aistls:requiresQueryProteinSequence>
```

この定義に基づいて PoodleL の入力 RDF を作成する。

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#"
  <aistls:PoodleLInput rdf:about="http://www.molprof.jp/ontologies/poodlel.rdf#1">
    <aistls:requiresQueryProteinSequence>>test
    MANLGCWMLVLFVATWSDLGLCKKRPKPGGWNTGGSRYPGQSPGGNRYPPQGGGGWGQPHGGGWGQPHGGGWGQPHGGGW
    GQPHGGGWGQGGGTHSQWNKPSKPKTNMKHMAGAAAAGAVVGGGLGGYMLGSAMSRPIIHFGSDYEDRRYRENMHRYPNQVYYRPMDE
    YSNQNNFVHDCVNITIKQHTVTTTTKGENFTETDVKMMERVVEQMCITQYERESQAYYQRGSSMVLFFSSPPVILLISFLIFLIVGMA
  </aistls:requiresQueryProteinSequence>
  </aistls:PoodleLInput>
</rdf:RDF>
```

**PoodleL 用入力 RDF**

### 2.3.2. 実行コマンド

---

利用するマシンから以下のコマンドを実行する。

```
% curl --data-binary @入力 RDF 名
http://semantic.medals.jp:8090/sadi-services/PoodleL
```

PoodleL は非同期通信となるため、まずポーリングに必要な URI を格納した RDF が返却される。次に、この RDF の `isDefinedBy` で定義された”`http://semantic.medals.jp:8090/sadi-services/PoodleL?poll=任意の 16 進数`”を用いて SADI サーバへのポーリングを行う。

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#"
  <aistls:PoodleLOutput rdf:about="http://www.molprof.jp/ontologies/poodlel.rdf#1">
    <rdfs:isDefinedBy
rdf:resource="http://semantic.medals.jp:8090/sadi-services/PoodleL?poll=21"/>
    </aistls:PoodleLOutput>
</rdf:RDF>
```

#### ポーリングのための URI を格納した RDF

```
% curl http://semantic.medals.jp:8090/sadi-services/PoodleL?poll=21 (上図の場合 21)
%
```

PoodleL が終了すると、PoodleL の実行結果を格納した URL が標準出力に返却される。

```
% curl http://semantic.medals.jp:8090/sadi-services/PoodleL?poll=21
COMPLETE: http://semantic.medals.jp/tmp/xxx/yyyy/poodleLResult.rdf
%
```

出力された URL を用いて、以下のコマンドで結果を取得する。

```
% curl http://semantic.medals.jp/tmp/xxx/yyyy/ poodleLResult.rdf -o 出力 RDF 名
```

### 2.3.3. 実行結果

---

以下の定義に従い、PoodleL の実行結果が RDF 形式で出力される。

- RDF のヘッダは以下となる。

```
<rdf:RDF
```

```
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#"
```

- 主語は PoodleLOutput クラス、rdf:about 属性は入力 RDF ファイルで指定したものがそのまま指定される。

```
<aistls:PoodleLOutput
```

```
  rdf:about="http://www.molprof.jp/ontologies/poodlel.rdf#1">
```

- PoodleL の実行結果のトリプルは以下となる。

```
  主語 : PoodleLOutput
```

```
  述語 : requiresResultInTextFormat
```

```
  目的語 : 文字列(PoodleL の実行結果)
```

```
<aistls:requiresResultInTextFormat>PoodleL 実行結果
```

```
</aistls:requiresResultInTextFormat>
```

PoodleL の出力結果の例を以下に示す。

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#"
  <aistls:PoodleLOutput rdf:about="http://www.molprof.jp/ontologies/poodlel.rdf#1">
    <aistls:requiresResultInTextFormat>PFRMAT DR
    REMARK S. Hirose, K. Shimizu, S. Kanai, Y. Kuroda and T. Noguchi
    ....
    A O 0.1241
    P O 0.2739
    END
    .....</aistls:requiresResultInTextFormat>
  </aistls:PoodleLOutput>
</rdf:RDF>
```

**PoodleL 出力結果**

## 2.4. PoodleS

---

### 2.4.1. 入力 RDF の準備

---

以下の方法で PoodleS の入力 RDF を作成する。

- RDF のヘッダに使用語彙 RDF、AISTLS SIO OWL を以下のように定義する。

```
<rdf:RDF
```

```
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#"
```

- 主語は PoodleSInput クラス、rdf:about 属性は任意とする。

```
<aistls:PoodleSInput
```

```
  rdf:about="http://www.molprof.jp/ontologies/poodles.rdf#1">
```

- アミノ酸配列のトリプルを定義する。

```
  主語 : PoodleSInput
```

```
  述語 : requiresQueryProteinSequence
```

```
  目的語 : 文字列(アミノ酸配列)
```

```
<aistls:requiresQueryProteinSequence>
```

```
  アミノ酸配列
```

```
</aistls:requiresQueryProteinSequence>
```

この定義に基づいて PoodleS の入力 RDF を作成する。

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#"
  <aistls:PoodleSInput rdf:about="http://www.molprof.jp/ontologies/poodles.rdf#1">
    <aistls:requiresQueryProteinSequence>>test
    MANLGCWMLVLFVATWSDLGLCKKRPKPGGWNTGGSRYPGQSPGGNRYPPQGGGGWGQPHGGGWGQPHGGGWGQPHGGGW
    GQPHGGGWGQGGGTHSQWNKPSKPKTNMKHMAGAAAAGAVVGGGLGGYMLGSAMSRPIIHFGSDYEDDRYYRENMHRYPNQVYYRPMDE
    YSNQNPNFVHDCVNITIKQHTVTTTTKGENFTETDVKMMERVVE</aistls:requiresQueryProteinSequence>
  </aistls:PoodleSInput>
</rdf:RDF>
```

#### PoodleS 用入力 RDF

## 2.4.2. 実行コマンド

---

利用するマシンから以下のコマンドを実行する。

```
% curl --data-binary @入力 RDF 名  
http://semantic.medals.jp:8090/sadi-services/PoodleS
```

PoodleS は非同期通信となるため、まずポーリングに必要な URI を格納した RDF が返却される。次に、この RDF の isDefinedBy で定義された  
”http://semantic.medals.jp:8090/sadi-services/PoodleS?poll=任意の 16 進数”  
を用いて SADI サーバへのポーリングを行う。

```
<rdf:RDF  
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"  
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#">  
  <aistls:PoodleSOutput rdf:about="http://www.molprof.jp/ontologies/poodles.rdf#1">  
    <rdfs:isDefinedBy  
rdf:resource="http://semantic.medals.jp:8090/sadi-services/PoodleS?poll=3F"/>  
  </aistls:PoodleSOutput>  
</rdf:RDF>
```

### ポーリングのための URI を格納した RDF

```
% curl http://semantic.medals.jp:8090/sadi-services/PoodleS?poll=3F (上図の場合  
3F)  
%
```

PoodleS が終了すると、PoodleS の実行結果を格納した URL が標準出力に返却される。

```
% curl http://semantic.medals.jp:8090/sadi-services/PoodleS?poll=3F  
COMPLETE: http://semantic.medals.jp/tmp/xxx/yyyy/poodleSResult.rdf  
%
```

出力された URL を用いて、以下のコマンドで結果を取得する。

```
% curl http://semantic.medals.jp/tmp/xxx/yyyy/ poodleSResult.rdf -o 出力 RDF 名
```



### 2.4.3. 実行結果

以下の定義に従い、PoodleS の実行結果が RDF 形式で出力される。

- RDF のヘッダは以下となる。

```
<rdf:RDF
```

```
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#"
```

- 主語は PoodleSOutput クラス、rdf:about 属性は入力 RDF ファイルで指定されたものがそのまま指定される。

```
<aistls:PoodleSOutput
```

```
  rdf:about="http://www.molprof.jp/ontologies/poodles.rdf#1">
```

- PoodleS の実行結果のトリプルは以下となる。

```
  主語 : PoodleSOutput
```

```
  述語 : requiresResultInTextFormat
```

```
  目的語 : 文字列(PoodleS の実行結果)
```

```
<aistls:requiresResultInTextFormat>PoodleS 実行結果
```

```
</aistls:requiresResultInTextFormat>
```

PoodleS の出力結果の例を以下に示す。

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#"
  <aistls:PoodleSOutput rdf:about="http://www.molprof.jp/ontologies/poodles.rdf#1">
    <aistls:requiresResultInTextFormat>PFRMAT DR
    REMARK K. Shimizu, Y. Muraoka, S. Hirose, and T. Noguchi
    REMARK "Feature Selection Based on Physicochemical Properties o
    A D 0.575
    N D 0.505
    ....
    Pharma Design, Inc. </aistls:requiresResultInTextFormat>
  </aistls:PoodleSOutput>
</rdf:RDF>
```

#### PoodleS 出力結果



## 2.5. Rassie

---

### 2.5.1. 入力 RDF の準備

---

以下の方法で Rassie の入力 RDF を作成する。

- RDF のヘッダに使用語彙 RDF、AISTLS SIO OWL を以下のように定義する。

```
<rdf:RDF
```

```
  xmlns:rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls "http://www.molprof.jp/ontologies/aistlssio.owl#"
```

- 主語は RassieInput クラス、rdf:about 属性は任意とする。

```
<aistls:RassieInput rdf:about="http://www.molprof.jp/ontologies/rassie.rdf#1">
```

- RNA2 次構造のトリプルを定義する。

主語 : RassieInput

述語 : requiresRNASecondaryStructure

目的語 : 文字列(RNA2 次構造)

```
<aistls:requiresRNASecondaryStructure>RNA2 次構造
```

```
</aistls:requiresRNASecondaryStructure>
```

- コマンドラインオプションのトリプルを定義する。

主語 : RassieInput

述語 : hasOptions

目的語 : 文字列(コマンドラインオプション)

```
<aistls:hasOptions> -q 100 -ins 100 -clst -outclst 10 c-ins_hain
```

```
</aistls:hasOptions>
```

この定義に基づいて Rassie の入力 RDF を作成する。

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#">
  <aistls:RassieInput rdf:about="http://www.molprof.jp/ontologies/rassie.rdf#1">
    <aistls:requiresRNASecondaryStructure>>1CQ5
    GGCGUUUACCAGGUCAGGUCCGGAAGGAAGCAGCCAAGGCGCC
    ((((((...(((...(((...)))...)))...)))...))) (g=4,th=0.2)
  </aistls:requiresRNASecondaryStructure>
  <aistls:hasOptions> -q 100 -ins 100 -clst -outclst 10 -ins_chain</aistls:hasOptions>
</aistls:RassieInput>
</rdf:RDF>
```

## Rassie 用入力 RDF

## 2.5.2. 実行コマンド

---

利用するマシンから以下のコマンドを実行する。

```
% curl --data-binary @入力 RDF 名
http://semantic.medals.jp:8090/sadi-services/Rassie
```

Rassie は非同期通信となるため、まずポーリングに必要な URI を格納した RDF が返却される。次に、この RDF の isDefinedBy で定義された  
”http://semantic.medals.jp:8090/sadi-services/Rassie?poll=任意の 16 進数”  
を用いて SADI サーバへのポーリングを行う。

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#"
  <aistls:RassieOutput rdf:about="http://www.molprof.jp/ontologies/rassie.rdf#1">
    <rdfs:isDefinedBy
rdf:resource="http://semantic.medals.jp:8090/sadi-services/Rassie?poll=30"/>
    </aistls:RassieOutput>
</rdf:RDF>
```

### ポーリングへの URI を格納した RDF

```
% curl http://semantic.medals.jp:8090/sadi-services/Rassie?poll=30 (上図の場合 30)
COMPLETE: http://semantic.medals.jp/tmp/xxx/yyyy/rna3dResult.rdf
%
```

Rassie が終了すると、Rassie の実行結果を格納した URL が標準出力に返却される。

```
% curl http://semantic.medals.jp:8090/sadi-services/Rassie?poll=30
COMPLETE: http://semantic.medals.jp/tmp/xxx/yyyy/rna3dResult.rdf
%
```

出力された URL を用いて、以下のコマンドで結果を取得する。

```
% curl http://semantic.medals.jp/tmp/xxx/yyyy/ rna3dResult.rdf -o 出力 RDF 名
```



### 2.5.3. 実行結果

---

以下の定義に従い、Rassie の実行結果が RDF 形式で出力される。

- RDF のヘッダは以下となる。

```
<rdf:RDF
```

```
  xmlns:rdf “http://www.w3.org/1999/02/22-rdf-syntax-ns#”
```

```
  xmlns:aistls “http://www.molprof.jp/ontologies/aistlssio.owl#”>
```

- 主語は RassieOutput クラス、rdf:about 属性は入力 RDF ファイルで指定したものがそのまま指定される。

```
<aistls:RassieOutput
```

```
  rdf:about=”http://www.molprof.jp/ontologies/rassie.rdf#1”>
```

- Rassie の実行結果のトリプルは以下となる。

```
  主語 : RassieOutput
```

```
  述語 : requiresResultInTextFormat
```

```
  目的語 : 文字列(Rassie の実行結果)
```

```
<aistls:requiresResultInTextFormat>Rassie 実行結果
```

```
</aistls:requiresResultInTextFormat>
```

Rassie の出力結果の例を以下に示す。

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aistls="http://www.molprof.jp/ontologies/aistlssio.owl#"
  <aistls:RassieOutput rdf:about="http://www.molprof.jp/ontologies/rassie.rdf#1">
    <aistls:requiresResultInTextFormat>MODEL 1
      ATOM    1 P      G X  1      12.409 -19.191  9.334
      ATOM    2 O5'   G X  1      11.280 -20.184  8.788
      ATOM    3 C5'   G X  1      11.573 -21.021  7.704
      ATOM    4 C4'   G X  1      10.414 -21.968  7.500
      ATOM    5 C3'   G X  1      9.158  -21.309  7.084
      ATOM    6 O3'   G X  1      8.938  -21.336  5.719
      ATOM    7 O4'   G X  1      10.030 -22.284  8.834
      .....
    .....</aistls:requiresResultInTextFormat>
    <aistls:requiresResultInTextFormat>MODEL 2
      ATOM    1 P      G X  1      12.409 -19.191  9.334
      ATOM    2 O5'   G X  1      11.280 -20.184  8.788
      ATOM    3 C5'   G X  1      11.573 -21.021  7.704
      ATOM    4 C4'   G X  1      10.414 -21.968  7.500
      ATOM    5 C3'   G X  1      9.158  -21.309  7.084
      ATOM    6 O3'   G X  1      8.938  -21.336  5.719
      ATOM    7 O4'   G X  1      10.030 -22.284  8.834
      .....
    .....</aistls:requiresResultInTextFormat>
  </aistls:RassieOutput>
</rdf:RDF>
```

### Rassie 出力結果

## おわりに

---

ご質問やご意見は下記のメールアドレスにお願いいたします。

[workflow@molprof.jp](mailto:workflow@molprof.jp)

産業技術総合研究所 創薬分子プロファイリング研究センター 情報統合班では、利用者のご要望を積極的に取り入れ、より良いシステムにしていく計画です。

---

産業技術総合研究所  
創薬分子プロファイリング研究センター  
情報統合班 <http://togo.medals.jp>

〒135-0064 東京都江東区青海 2-4-7

産総研 臨海副都心センター別館 (バイオ・IT 融合研究棟)