

MassSpecBlocks

Databáze sekvencí a stavebních bloků mikrobiálních metabolitů
pro analýzu hmotnostních spekter

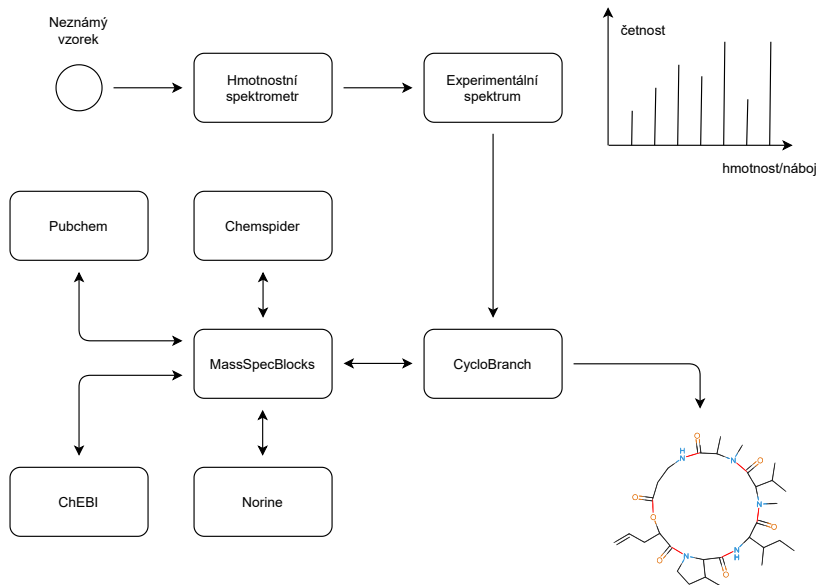
Jan Přívratský

30.05.2021

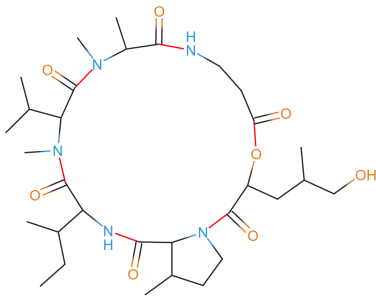
MassSpecBlocks

- 1 webová open-source aplikace, vychází z Bbdgnc
- 2 správa sekvencí a bloků pro hmotnostní analýzu
- 3 nově: uživatelé, podobnost, polyketidy, N-C směr, ...
- 4 backend + frontend + relační DB
- 5 PHP, Symfony + Typescript, React + MySQL (MariaDB)

Proces



MSB ukázka I.

[Login](#)

Success! Found more, select one

Notice - Create new sequence

Nonribosomal Peptides and Siderophores

Database

MSB - Nonribosomal Peptides and Siderophores

Search by

native

Nar

relative
rise of coan

SMILES

CCC(C)C1NC(=O)C2C(C)CCN2C(=O)C(CC(C)CO)OC(=O)CCNC(=O)C(C)N(C)C(=O)C(C(C)C)N(C)C1=O

Molecular Formula

C₃₁H₅₃N₅O₈

Monoisotopic Mass

623.389414

Identifier

845

Find

Edit

Generic SMILES

Unique SMILES

Build Blocks

Soyuz




MSB ukázka II.

Sequence - 6 blocks

Type: cyclic Sequence: [Ile]-[NMe-Val]-[NMe-Ala]-[Ala]-[C5:0-Me(4)-OH/2:5]-[3Me-Pro]

Family: lipidomics lipidomics x v Organism: Select... v

☒ Edit same blocks together

MSB acronym	Preview	Acronym	SMILES	Name	Formula	Mass	Losses	Identifier	Actions
Ile		Ile	<chem>CCC(C)(NC)C(=O)O</chem>	Isoleucine	C6H11NO	113.084064		CID: 6306	Edit Remove
NMe-Val		NMe-Val	<chem>CNC(C)C(=O)O</chem>	N-methyl-valine	C6H11NO	113.084064		CID: 4378	Edit Remove
NMe-Ala		NMe-Ala	<chem>CNC(C)C(=O)O</chem>	N-methyl-alanine	C4H7NO	85.052764		CID: 5288725	Edit Remove

MSB ukázka III.

Create new container

Container name: [Create new container](#)

Your containers - 1 rows

Container name ⇅	Visibility ⇅	Mode ⇅	Is selected	Actions				
My container	PRIVATE	RWM	Yes	Select	Details	Clone	Export	Delete

Public containers - 3 rows

Container Name ⇅	Is selected	Actions		
Nonribosomal Peptides and Siderophores	No	Select	Clone	Export
Proteinogenic Amino Acids	No	Select	Clone	Export
Siderophores and Secondary Metabolites (MS)	No	Select	Clone	Export

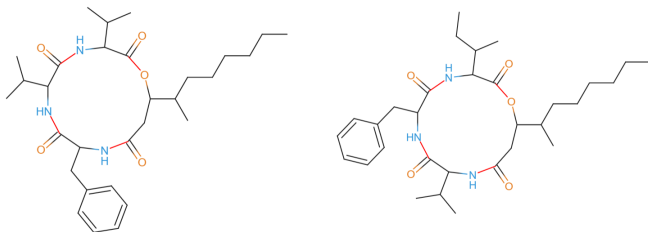
MSB ukázka IV.

[MassSpecBlocks](#)[Containers](#)[Sequences](#)[Blocks](#)[Modifications](#)[Login](#)

List of blocks - Proteinogenic Amino Acids - 20 rows

Name ↕	Acronym ^	Residue ↕	Mass ↕	Losses ↕	Family ↕	SMILES ↕	Identifier ↕	Actions
<input type="text" value="Name"/>	<input type="text" value="Acronym"/>	<input type="text" value="Formula"/>	<div><input type="text" value="Mass from"/> <input type="text" value="Mass to"/></div>	<input type="text" value="Losses"/>	<input type="text" value="Family"/>	<input type="text" value="Smiles"/>	<input type="text" value="Identifier"/>	Filter Clear
Alanine	Ala	C3H5NO	71.037114		proteinogenic amino acids	CC(N)C(O)=O	CID: 5950	Editor Show FindRef Clone Usage Delete
Arginine	Arg	C6H12N4O	156.101111		proteinogenic amino acids	NC(CCCN=C(N)N)C(O)=O	CID: 6322	Editor Show FindRef Clone Usage Delete
Asparagine	Asn	C4H6N2O2	114.042927		proteinogenic amino acids	NC(CC(N)=O)C(O)=O	CID: 6267	Editor Show FindRef Clone Usage Delete

Tanimoto - podobnost sekvencí I.



1 BeauverolideA = {Val, Phe, C10:0-Me(4)-OH(3)} (2x Val)

2 BeauverolideB = {Val, Phe, Ile, C10:0-Me(4)-OH(3)}

3
$$T = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} = \frac{3}{3 + 4 - 3} = \frac{3}{4} = 0.75$$

Tanimoto - podobnost sekvencí II.

	SEQUENCE_ID	SEQUENCE_NAME	TANIMOTO	RN
	24	beauverolide A	1.0000	1
	25	beauverolide B	0.7500	2
	779	beauverolide Ba	0.7500	3
	26	beauverolide C	0.5000	4
	27	beauverolide D	0.5000	5
	28	beauverolide E	0.4000	6
	780	beauverolide Ea	0.4000	7
	775	beauverolide II	0.4000	8
	777	beauverolide Ka	0.4000	9
	31	beauverolide L	0.4000	10
	776	beauverolide La	0.4000	11
	791	beauverolide P	0.4000	12
	786	beauverolide VII	0.4000	13
	787	beauverolide VIII	0.4000	14
	33	gramicidin S	0.3333	15
	20	pseudacyclin B	0.2857	16

Závěr

- 1 webová uživatelská aplikace
- 2 správa chemických struktur a jejich vyhledávání
- 3 využití v hmotnostní analýze

Odpovědi na otázky

- 1 Proč jste zvolil adresy API s `/rest` namísto běžnějšího `/api`? Čekáte, že backend bude využíván jinými aplikacemi nebo jen Vaším frontendem?
- 2 V práci zmiňujete paralelizaci, jakým způsobem je jí dosaženo (na úrovni kódu)? Je využit preemptivní nebo kooperativní multitasking? V čem tkví zrychlení pomocí paralelizace (čekání na odpověď z externího API, zpracování odpovědi, ...)?
- 3 U aplikace SmilesDrawer (dříve vytvořená aplikace autorem ZP, použitá novou aplikací) se z historických důvodů používá Travis CI. Nebylo by lepší zmigrovat také na GitHub Actions z pohledu udržitelnosti? Jaká by byla náročnost či případné překážky?

Odpovědi na otázky - API

Souhlasím, že použít /api by bylo vhodnější. Nepřišlo mi to důležité, pro zajímavost jsem se koukl na URL API, která používám a ani jedno /api nepoužívá, většina používá /rest/v1 a pod. Takto tam již vniká rozpor mezi /api/doc a /rest.

Zatím nepředpokládám, ale připravujeme ještě s vedoucím článek pro chemický časopis, kde by se mohlo zvýšit povědomí o aplikaci, takže by se někdo najít mohl. Preferovanou variantou by bylo nasadit si svůj vlastní backend a používat ten.

Odpovědi na otázky - paralelizace I.

JavaScript je single-thread, tedy není preemptivní ani kooperativní multitasking. Použil jsem knihovnu `async-parallel`, která uvnitř používá `async-await` nad polem. Knihovna v metodě `pool()` managuje počet "vláken" - vytvoří tolik asynchronních volání kolik je specifikováno a pokud je ještě další práce tak vyvolá další asynchroní volání uvnitř původního asynchronního volání. Zrychlení je především v čekání na odpovědi z externího API.

Odpovědi na otázky - paralelizace II.

```
125  /**
126   * Creates a new array with the results of calling a provided function in parallel on every input.
127   * The output will be in the same order as the input.
128   * @param list A list of input elements to map.
129   * @param action An async function callback that produces an element of the output list. The callback takes three arguments.
130   * @param options Limits the number of callback actions to run concurrently.
131   * @returns A list of mapped elements in the same order as the input.
132   */
133  export async function map<T1, T2>(list: T1[], action: {(value: T1, index: number, list: T1[]): Promise<T2>}, options?: Options): Promise<T2[]> {
134      var result: T2[] = [];
135      if (list && list.length > 0) {
136          list = list.slice(0);
137          var size = resolveOptions(options).concurrency || list.length;
138          var i = 0;
139          await pool(size, async () => {
140              if (list.length > 0) {
141                  var j = i++;
142                  result[j] = await action(list.shift()!, j, list);
143              }
144              return list.length > 0;
145          });
146      }
147      return result;
148  }
```

Odpovědi na otázky - paralelizace III.

```
150  /**
151   * Repeatedly invokes a provided async function until `false` is returned, after which no new instances will be invoked.
152   * The overall operation is resolved when all existing instances have been resolved.
153   * @param size Specifies the size of the pool indicating the number of parallel instances of the provided async function to maintain.
154   * @param task The provided async function callback that takes no arguments and resolves to a boolean. Return `true` to continue, or `false` to stop.
155   */
156  export async function pool(size: number, task: (() : Promise<boolean>)): Promise<void> {
157      var active = 0;
158      var done = false;
159      var errors: Array<Error> = [];
160      return new Promise<void>((resolve, reject) => {
161          next();
162          function next(): void {
163              while (active < size && !done) {
164                  active += 1;
165                  task()
166                      .then(more => {
167                          if (--active === 0 && (done || !more))
168                              errors.length === 0 ? resolve() : reject(new MultiError(errors));
169                          else if (more)
170                              next();
171                          else
172                              done = true;
173                      })
174              }
175          }
176      })
177  }
```

Odpovědi na otázky - paralelizace IV.

Name	Status	Type	Initiator	Size	Time	Waterfall
<input type="checkbox"/> smiles	200	fetch	MainPage.tsx:674	1.7 kB	52 ms	
<input type="checkbox"/> json?list_return=listkey	200	fetch	PubChemFinder.ts:128	501 B	1.07 s	
<input type="checkbox"/> json?list_return=listkey	200	fetch	PubChemFinder.ts:128	759 B	1.03 s	
<input type="checkbox"/> sim	400	fetch	MainPage.tsx:739	348 B	32 ms	
<input type="checkbox"/> json?listkey=279274...	200	fetch	PubChemFinder.ts:133	266 B	475 ms	
<input type="checkbox"/> json?listkey=396252...	200	fetch	PubChemFinder.ts:133	314 B	438 ms	
<input type="checkbox"/> json	200	fetch	PubChemFinder.ts:63	489 B	224 ms	
<input type="checkbox"/> json	200	fetch	PubChemFinder.ts:63	440 B	225 ms	

Odpovědi na otázky - SmilesDrawer CI

S Travis CI nebyly žádné velké problémy tak mě nenapadlo přejít na GitHub Actions. Z hlediska konzistence a udržitelnosti nevidím důvod proč nemigrovat. Bylo by potřeba připravit script, který provede build a spustí testy. Překážky nevidím a časová náročnost bude nízká (max 2h s problémy, když to půjde ideálně 30min).