

# FungiRegEx Software documentation



This guide can be useful if you want to use the software.

**Software Name:** FungiRegEx

**Version:** 1.0.0

**Date:** 📅 Aug 10, 2022

## TABLE OF CONTENTS

- Previous steps
- Software summary
- Requirements 🙌
- Initial configuration ⚙️
  - Local server/local computer/external server
  - Own computer
- Cluster configuration ⚙️
- About Regular Expressions 🖨️
- How-to guide 🐥
  - Step 1 - Opening consoles and execute commands to start Web based applic...
  - Step 3 - Open the Web-based application
  - Step 4 - Go and use FungiRegEx
  - Step 5 - Write the Regular Expression
  - Step 6 - Search
- Tutorials 🎬
- FAQs 🙋

- Question
- Question
- Question

---

- Additional Resources 

---

- Support 

---

## Previous steps

Before executing the software you have to run the following commands and make sure that have all the prerequisites:

1. You receive the complete folder with the project and the code.
2. First is needed to install all the dependencies, to perform this you have to run in the folder execute:

```
1  npm i
```

3. Then install nodemon:

```
1  npm install -g nodemon
```

4. Then install chromium, depending the OS, check documentation:

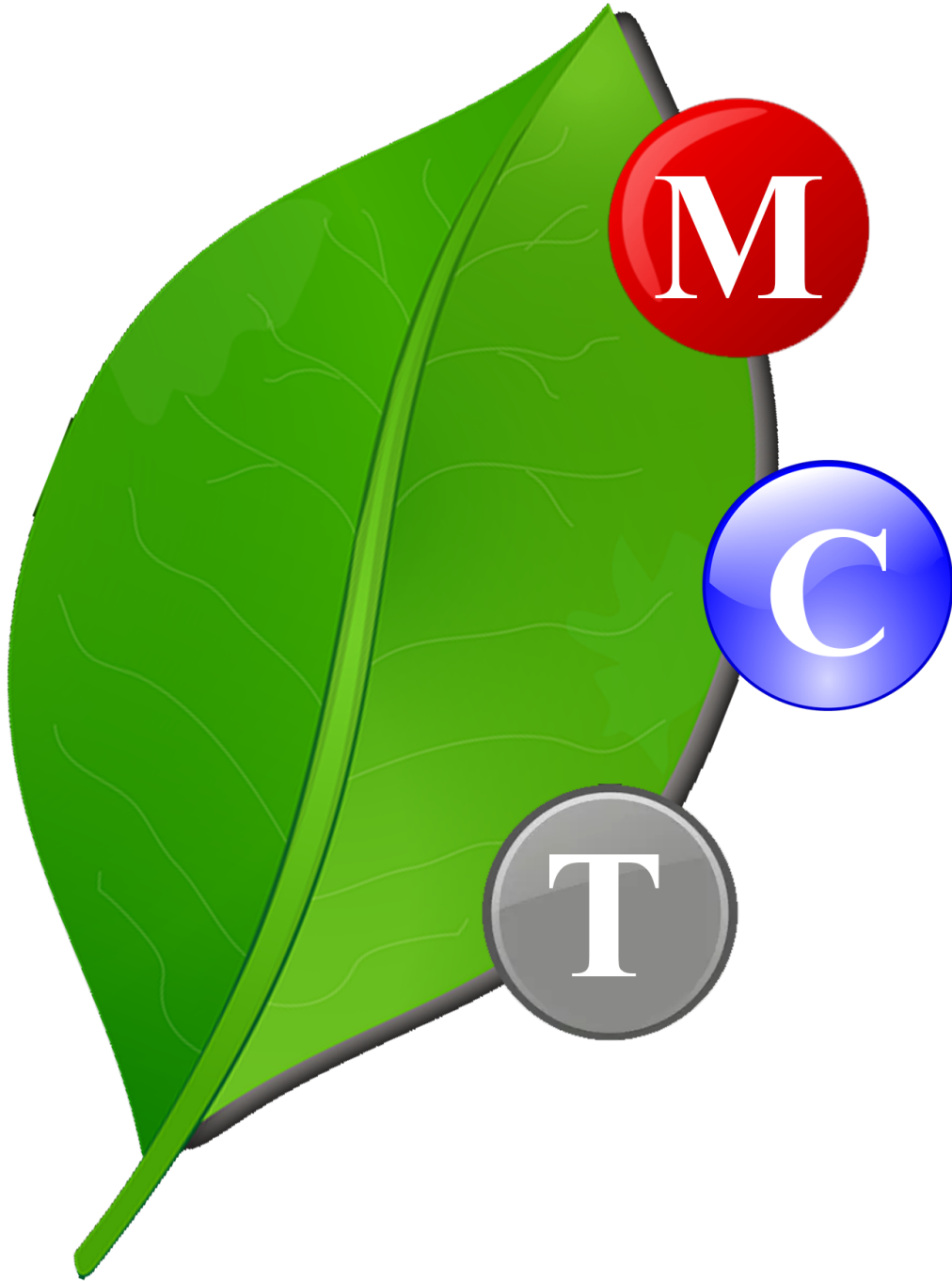
```
1  choco install googlechrome --version=79.0.3945.117
```

After performing the previous steps you are ready to execute front-end and backend commands.

## Software summary

This tool is a web-based search engine for regular expressions in the proteomes, all the information is obtained from the JGI (Joint Genome Institute) database through a scraper for all the available species; therefore this tool only considers fungi organisms.

In this version, we use React JS in front-end and NodeJS + Express for back-end.



It is also important to note that the number of chromium instances the computer can open in parallel is configurable based on the computer's resources.

## Requirements 🙌

This software has been developed and tested using a computer with the next characteristics:

- 4 GB RAM
- Cent OS 7 and Windows 10
- Core i7 5<sup>th</sup> generation processor

About the characteristics of software:

- Node JS 16.17.0v
- Chromium 79.0.3945.117v
- React JS 17.0.2v

The project contains all the libraries needed to deploy it, just write the next lines in the bash or console according to your OS needed that you are in the directory where you download the application.

- `npm run start:frontend`
- `npm run start:backend`

## Initial configuration

You can deploy this application both on a local server/computer and on your own computer, this will be determined by each user depending on the available computing resources and the availability that each user wants.

Take into account that if multiple users connect at the same time, they will interrupt the tasks that are being executed and the new user will be given priority, that is, the tasks of the previous user will be eliminated from the queue and the tasks of the new user will be executed.

## Local server/local computer/external server

In case you deploy the application using a local server (which can be another computer) follow the next instructions:

1. Get the IP address of the server/local server/computer.
  - a. For windows you can follow the next instructions:
    - i. <https://support.microsoft.com/en-us/windows/find-your-ip-address-in-windows-f21a9bbc-c582-55cd-35e0-73431160a1b9>
  - b. For LINUX:
    - i. Ubuntu: <https://help.ubuntu.com/stable/ubuntu-help/net-findip.html.en>

- ii. Cent OS / Red Hat: [https://access.redhat.com/documentation/es-es/red\\_hat\\_enterprise\\_linux/7/html/migration\\_planning\\_guide/ch04s04s07](https://access.redhat.com/documentation/es-es/red_hat_enterprise_linux/7/html/migration_planning_guide/ch04s04s07)
  - iii. For other distributions look for information on official websites.
2. Navigate to the filesystem until you find the folder that contains the application.
  3. As the applications run in an external computer different from the user we need to change the IP address parameters in the next files:

```
1  ../../interfacebio/src/components/main/Main.jsx
2  ../../interfacebio/src/components/table/Table.jsx
```

For the Main.jsx file you have to modify the IP address on lines: 14, 117 and 15, for example, let's suppose that the external computer/server has the IP 555.811.22.33 the code that you will find in the application is:

```
1  const [products, setProducts] = useState([])
2  const [progressVal, setProgressVal] = useState(0)
3  const [blocked, setBlocked] = useState(false);
4  const endpoint = "http://192.168.1.102:8000/file" //line 14
5  const progress = "http://192.168.1.102:8000/progress"//line
  15
```

You have to change IP address values to:

```
1  const [products, setProducts] = useState([])
2  const [progressVal, setProgressVal] = useState(0)
3  const [blocked, setBlocked] = useState(false);
4  const endpoint = "http://555.811.22.33:8000/file"//line 14
5  const progress = "http://555.811.22.33:8000/progress"//line
  15
```

In line 117 you will find:

```
1  try {
2      await
    axios.post("http://192.168.1.102:8000/resultsScrap", {
3      stringtoBackend
4      })
5
6  }
```

You have to change the IP address value to:

```

1  try {
2      await
    axios.post("http://555.811.22.33:8000/resultsScrap", {
3          stringtoBackend
4      })
5
6  }

```

For the file Table.jsx you have to modify the IP address on lines: 10, for example, let's suppose that the external computer/server has the IP 555.811.22.33 the code that you will find in the application is:

```

1  const endpoint = "http://192.168.1.102:3000/listaURLnew.json"

```

You have to change the IP address value to:

```

1  const endpoint = "http://555.811.22.33:3000/listaURLnew.json"

```

## Own computer

You have to get the IP address of your computer, you can use the last section steps from the last section.

Then also edit the files and change the IP value with the IP address of your computer.

## Cluster configuration

The principle of operation of the application is based on opening multiple tabs of the chromium browser, obtaining the list of URLs, and making requests to the JGI server. The number of tabs you open on the computer where it is run depends on this parameter and will determine the amount of resources to use.

Considering the previous requirements we use no more than 50 instances. You can set this parameter by opening the file

`directory_where_you_download_the_application/interfacebio/backend.js`

In this file you will see this part of the code:

```

1  var cluster = await Cluster.launch({
2      concurrency: Cluster.CONCURRENCY_CONTEXT,

```

```

3      maxConcurrency: 30,
4      puppeteerOptions: {
5          headless: true,
6          defaultViewport: false,
7          args: ['--no-sandbox', '--disable-setuid-sandbox'],
8
9      },
10     });

```

maxConcurrency: 30. This is the parameter to modify (no more than 50 is recommended). The higher the number placed, the higher the search speed and the higher the consumption of resources.

## About Regular Expressions

To match a single "a" followed by zero or more "b"s followed by "c", you'd use the pattern `/ab*c/`; the `*` after "b" means "0 or more occurrences of the preceding item." In the string `"cbbabbbbbcdebc"`, this pattern will match the substring `"abbbbc"`.

If you need to use any of the special characters literally (actually searching for a `"*"`, for instance), you must escape it by putting a backslash in front of it. For instance, to search for "a" followed by "\*" followed by "b", you'd use `/a\\*b/` — the backslash "escapes" the `"*"`, making it literal instead of special.

For more information about Regular Expressions see: [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular\\_Expressions](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_Expressions)

Just remember that you don't have to write `/` at the beginning and at the end, just place the amino acids that you want to search for.

1	Characters	
2	x	The character x
3	\\	The backslash character
4	\\0n	The character with octal value 0n (0 <= n <= 7)
5	\\0nn	The character with octal value 0nn (0 <= n <= 7)
6	\\0mnn	The character with octal value 0mnn (0 <= m <= 3, 0 <= n <= 7)
7	\\xhh	The character with hexadecimal value 0xhh
8	\\uhhhh	The character with hexadecimal value 0xhhhh
9	\\t	The tab character ('\\u0009')

```

10  \n      The newline (line feed) character ('\u000A')
11  \r      The carriage-return character ('\u000D')
12  \f      The form-feed character ('\u000C')
13  \a      The alert (bell) character ('\u0007')
14  \e      The escape character ('\u001B')
15  \cx     The control character corresponding to x
16
17  Character classes
18  [abc]    a, b, or c (simple class)
19  [^abc]   Any character except a, b, or c (negation)
20  [a-zA-Z] a through z or A through Z, inclusive (range)
21  [a-d[m-p]] a through d, or m through p: [a-dm-p] (union)
22  [a-z&&[def]] d, e, or f (intersection)
23  [a-z&&[^bc]] a through z, except for b and c: [ad-z]
    (subtraction)
24  [a-z&&[^m-p]] a through z, and not m through p: [a-lq-z]
    (subtraction)
25
26  Predefined character classes
27  .        Any character (may or may not match line terminators)
28  \d       A digit: [0-9]
29  \D       A non-digit: [^0-9]
30  \s       A whitespace character: [ \t\n\x0B\f\r]
31  \S       A non-whitespace character: [^\s]
32  \w       A word character: [a-zA-Z_0-9]
33  \W       A non-word character: [^\w]
34  Boundary matchers
35  ^        The beginning of a line
36  $        The end of a line
37  \b       A word boundary
38  \B       A non-word boundary
39  \A       The beginning of the input
40  \G       The end of the previous match
41  \Z       The end of the input but for the final terminator, if any
42  \z       The end of the input
43
44  Greedy quantifiers
45  X?       X, once or not at all
46  X*       X, zero or more times
47  X+       X, one or more times

```



```

48  X{n}      X, exactly n times
49  X{n,}    X, at least n times
50  X{n,m}   X, at least n but not more than m times
51
52  Reluctant quantifiers
53  X??      X, once or not at all
54  X*?      X, zero or more times
55  X+?      X, one or more times
56  X{n}?    X, exactly n times
57  X{n,}?   X, at least n times
58  X{n,m}?  X, at least n but not more than m times
59
60  Possessive quantifiers
61  X?+      X, once or not at all
62  X*+      X, zero or more times
63  X++      X, one or more times
64  X{n}+    X, exactly n times
65  X{n,}+   X, at least n times
66  X{n,m}+  X, at least n but not more than m times
67
68  Logical operators
69  XY       X followed by Y
70  X|Y      Either X or Y

```

## How-to guide 🐤

Before running it, make sure that you have the following installed:

- Node JS 16.17.0v
- Chromium browser in a stable version

If you don't have NodeJS installed you can get it from: <https://nodejs.org/en/>

If you don't have chromium installed you can get it from:

<https://www.chromium.org/getting-involved/download-chromium/>

Just consider the OS.

## Step 1 - Opening consoles and execute commands to start Web based application

Execute the next commands to start the application, in the first console:

```
1 npm run start:frontend
```

You can see the output of this command in Figure 1.

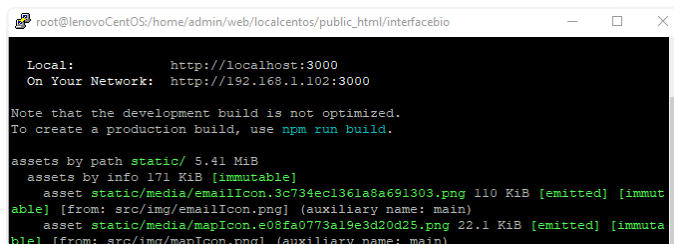
In the second console, run the next command:

```
1 npm run start:backend
```

You can see the output of this command in Figure 2.

To stop the app just hit Ctrl+C on the consoles.

Open Console or bash (2 instances):

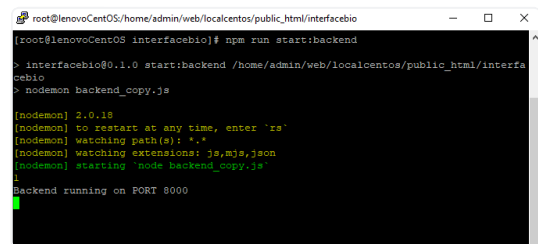


```
root@lenovoCentOS:/home/admin/web/localcentos/public_html/interfacebio
Local:      http://localhost:3000
On Your Network:  http://192.168.1.102:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

assets by path static/ 5.41 MiB
  assets by info 171 KiB [immutable]
    asset static/media/emailIcon.3c734ec1361a8a691303.png 110 KiB [emitted] [immutable] [from: src/img/emailIcon.png] (auxiliary name: main)
    asset static/media/mapIcon.e08fa0773a19e3d20d25.png 22.1 KiB [emitted] [immutable] [from: src/img/mapIcon.png] (auxiliary name: main)
```

Figure 1. Executing front-end



```
root@lenovoCentOS:/home/admin/web/localcentos/public_html/interfacebio
root@lenovoCentOS interfacebio# npm run start:backend
interfacebio@0.1.0 start:backend /home/admin/web/localcentos/public_html/interfacebio
> nodeemon backend_copy.js

(nodeemon) 2.0.18
(nodeemon) to restart at any time, enter 'rs'
(nodeemon) watching path(s): *.*
(nodeemon) watching extensions: js,mjs,json
(nodeemon) starting 'node backend_copy.js'
^I
Backend running on PORT 8000
```

Figure 2. Executing back-end

Is important to see the output cause it indicates if an error during execution occurs. Note: the back-end uses a not optimized React App, if you want to deploy an optimized application you can execute the next command:

```
1 npm run build
```

This command will create an optimized version.

### Step 3 - Open the Web-based application

After steps 1 and 2 you can go to your browser, and open localhost:3000 or the address that indicates the console window.

In Figure 1 we see that console indicates that the address is <http://localhost:3000> or <http://192.168.1.102:3000> but it could variate so just make sure that you have the right address, then place it into the navbar of the browser.

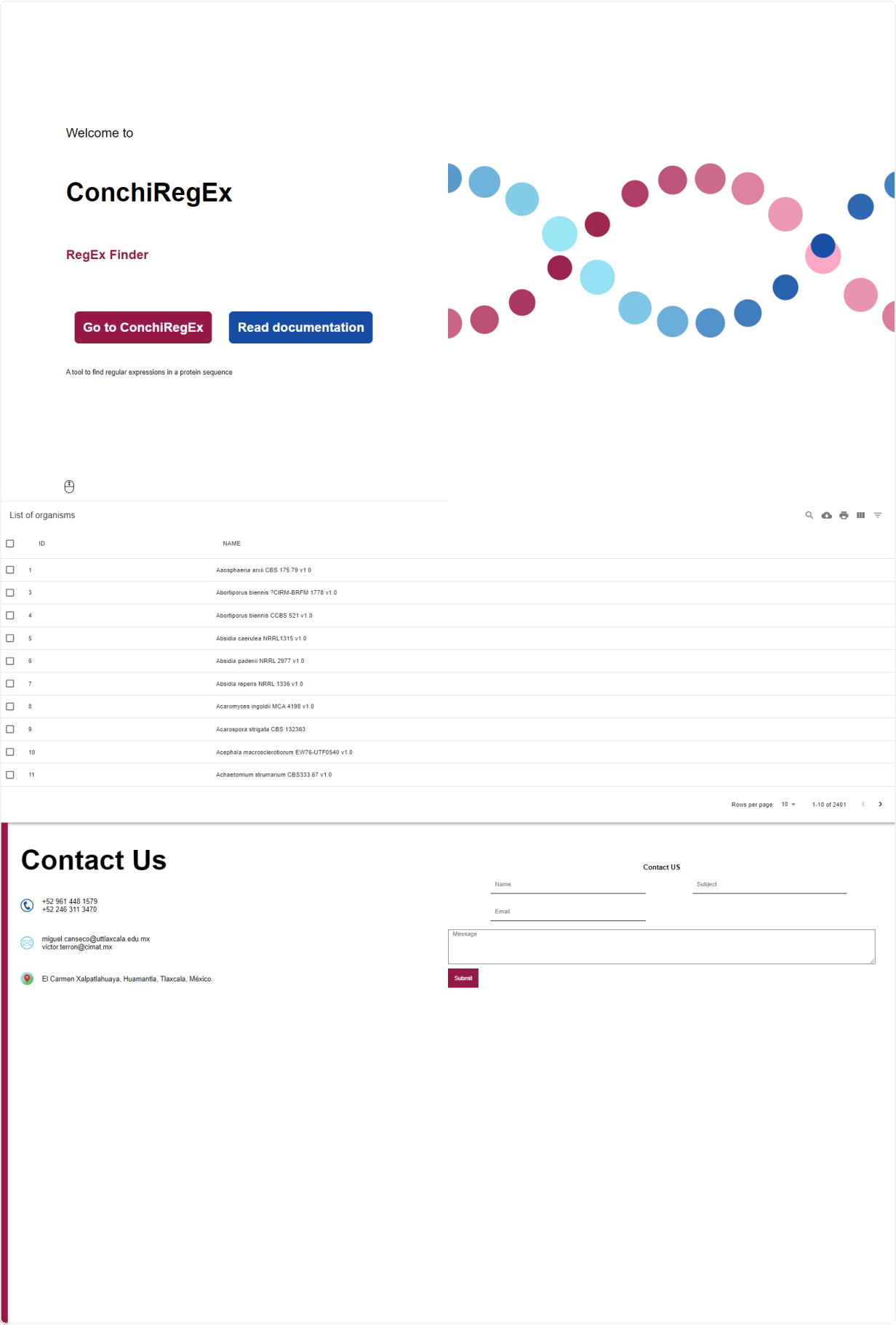


Figure 3. Interface of FungiRegEx

We see in Figure 3 the interface consists of 3 main components:

- Introduction
- Table of supported species
- A form to contact us

You can search for the scientific name of a specie, download the table data or filter the species.

## Step 4 - Go and use FungiRegEx

After steps 1 and 2 you can go to your browser, and open localhost:3000 or the address that indicates the console window.

The screenshot displays the 'Fill the required information' section of the FungiRegEx application. It includes a 'CHECK CONFIGURATIONS.' message and instructions on how to perform a global search or a specific search. There are four radio buttons for search types: 'SPECIFIC SPECIE', 'GLOBALLY', 'SPECIFIC RANGE', and 'LIST OF IDS'. The 'GLOBALLY' option is selected. Below these options is a 'Regular expression:' label and a text input field containing 'FAA\*'. A 'RETURN TO HOMEPAGE' button is located in the top right corner. On the right side, there is a 'List of results' table with columns: SPECIE, ID, VERSION, PROTEOM, REGULAR\_EXPRESSION, # MATCHES, and MATCHES. The table currently shows 'Sorry, no matching records found' and a pagination bar indicating 'Rows per page: 10' and '0-0 of 0'.

SPECIE	ID	VERSION	PROTEOM	REGULAR_EXPRESSION	# MATCHES	MATCHES
Sorry, no matching records found						

Figure 4. Main interface

The main interface consists of:

- Type of search:
  - Globally: if you want to search in all the species and look for Regular Expression

## Fill the required information

CHECK CONFIGURATIONS.

Select if you want to perform a global search or look into specific specie. Take in consideration that it will take a lot of time cause the application is able to search in 2,402 different species.

☐ SPECIFIC SPECIE

☒ GLOBALLY

Select if you want to scrap in a range of IDs or if you want to scan a specific list of IDs.

☐ SPECIFIC RANGE

☐ LIST OF IDS

From:  Regular expression:

To:

SCRAP

RETURN TO HOMEPAGE

List of results						
SPECIE	ID	VERSION	PROTEOM	REGULAR_EXPRESSION	# MATCHES	MATCHES
Sorry, no matching records found						
Rows per page: 10 0-0 of 0 < >						

Figure 5. Search Globally

- Specific species: If you want to search in the proteome of a specific specie the Regular Expression

## Fill the required information

CHECK CONFIGURATIONS.

Select if you want to perform a global search or look into specific specie. Take in consideration that it will take a lot of time cause the application is able to search in 2,402 different species.

☒ SPECIFIC SPECIE

☐ GLOBALLY

Aaospaeria arxii CBS 175.79 v1.0  
Aaospaeria arxii CBS 175.79 v1.0  
Abortiporus biennis ?CIRM-BRFM 1778 v1.0  
Abortiporus biennis CCBS 521 v1.0  
Absidia caerulea NRRL1315 v1.0  
Absidia padenii NRRL 2977 v1.0  
Absidia repens NRRL 1336 v1.0  
Acaromyces ingoldii MCA 4198 v1.0  
Acarospora strigata CBS 132363  
Acephala macrosclerotium EW76-UTF0540 v1.0  
Achaetomium strumarium CBS333.67 v1.0  
Acidomyces richmondensis BFW  
Acidothrix acidophila CBS 136259 v1.0  
Acremonium alcalophilum v2.0  
Acremonium chrysogenum ATCC 11550  
Acremonium strictum DS1bioAY4a v1.0  
Acrocalymma vagum DSE9081 v1.0  
Agaricostilbum hyphaenes ATCC MYA-4628 v1.0  
Agaricostilbum hyphaenes ATCC MYA-4628 v1.0  
Agaricus bisporus var bisporus (H97) v2.0

RETURN TO HOMEPAGE

List of results						
SPECIE	ID	VERSION	PROTEOM	REGULAR_EXPRESSION	# MATCHES	MATCHES
Sorry, no matching records found						
Rows per page: 10 0-0 of 0 < >						

Figure 6. Specific Specie selector

- Range or List of ID:
  - You can set a range for the tool to search in.

## Fill the required information

CHECK CONFIGURATIONS.

Select if you want to perform a global search or look into specific specie. Take in consideration that it will take a lot of time cause the application is able to search in 2,402 different species.

☐ SPECIFIC SPECIE

☒ GLOBALLY

Select if you want to scrap in a range of IDs or if you want to scan a specific list of IDs.

☒ SPECIFIC RANGE

☐ LIST OF IDS

From:  Regular expression:

To:

SCRAP

RETURN TO HOMEPAGE

List of results

🔍 📄 🖨️ 📊 🏠

SPECIE	ID	VERSION	PROTEOM	REGULAR_EXPRESSION	# MATCHES	MATCHES
--------	----	---------	---------	--------------------	-----------	---------

Sorry, no matching records found

Rows per page: 10 0-0 of 0 < >

Figure 7. Range

In this version, we include 2,402 different species, available in the selector.

## Step 5 - Write the Regular Expression

This tool is characterized by the ease of writing regular expressions since they are written using the sequence that the user wants. To learn more about supported regular expressions you can consult the specific section "about regular expressions".

For example, to match a single "FA" followed by zero or more "A"s, you'd use the pattern `FAA*`: the `*` after "A" means "0 or more occurrences of the preceding item."

Regular expression:

FAA\*

Figure 8. Regular Expression

## Step 6 - Search

After all the previous steps you can click the search button, if you did not perform the steps the app will not perform anything.

Also, consider that at the beginning the application can take a while, as soon as the results are obtained they will be displayed in the table.

The speed of search depends on different factors:

- Load of servers of JGI
- Internet speed of the user
- Computer resources
- Configuration of the cluster to open multiple instances, this parameter is available in the backend.js file, to modify this parameter see the section "cluster configuration"

For this example, I perform a search from 1 to 1000 ID in *Trichoderma aethiopicum* CBS130628 v1.0 using the regular expression `FAA*`. The results are:

The results will have the next order:

- Specie
- ID
- Scaffold / Version
- Proteome
- #matches
- Matches

If the query to the JGI server with the species and the identifier is empty, that is, JGI does not have the proteome, `NO_SHORTNAME`, `NO DATA` will be displayed.

### Fill the required information

CHECK CONFIGURATIONS.

Select if you want to perform a global search or look into specific specie. Take in consideration that it will take a lot of time cause the application is able to search in 2.402 different species.

☒ SPECIFIC SPECIE

☐ GLOBALLY

Select if you want to scrap in a range of IDs or if you want to scan a specific list of IDs.

☒ SPECIFIC RANGE

☐ LIST OF IDS

From:  To:

**PROGRESS**  
26.4 %

[RETURN TO HOMEPAGE](#)

SPECIE	ID	VERSION	PROTEOM	REGULAR_EXPRESSION	# MATCHES	MATCHES
Triaet1	13	NO_SHORTNAME	NO_DATA	FAA*	0	NO MATCHES
Triaet1	5	NO_SHORTNAME	NO_DATA	FAA*	0	NO MATCHES
Triaet1	15	NO_SHORTNAME	NO_DATA	FAA*	0	NO MATCHES
Triaet1	25	NO_SHORTNAME	NO_DATA	FAA*	0	NO MATCHES
Triaet1	17	NO_SHORTNAME	NO_DATA	FAA*	0	NO MATCHES
Triaet1	8	NO_SHORTNAME	NO_DATA	FAA*	0	NO MATCHES
Triaet1	3	NO_SHORTNAME	NO_DATA	FAA*	0	NO MATCHES
Triaet1	27	NO_SHORTNAME	NO_DATA	FAA*	0	NO MATCHES
Triaet1	20	NO_SHORTNAME	NO_DATA	FAA*	0	NO MATCHES
Triaet1	28	NO_SHORTNAME	NO_DATA	FAA*	0	NO MATCHES

Rows per page: 10 1-10 of 265

Figure 9. Results

Protein	Accession	Score	Rank
Protein 1	Accession 1	Score 1	Rank 1
Protein 2	Accession 2	Score 2	Rank 2
Protein 3	Accession 3	Score 3	Rank 3
Protein 4	Accession 4	Score 4	Rank 4
Protein 5	Accession 5	Score 5	Rank 5
Protein 6	Accession 6	Score 6	Rank 6
Protein 7	Accession 7	Score 7	Rank 7
Protein 8	Accession 8	Score 8	Rank 8
Protein 9	Accession 9	Score 9	Rank 9
Protein 10	Accession 10	Score 10	Rank 10

Figure 10. Results

During the search process, the results will be displayed in the table, you can order by the number of matches, also the table will contain the proteome.

Take into account that requests to the JGI server sometimes do not return anything, so if the result of the request is empty, the application will not show this information in the table. Given this situation, in the previous search of 1000 identifiers that were passed as search parameters, only 995 were obtained. It is also clarified that this does not happen frequently and that it is unrelated to the operation of this tool.








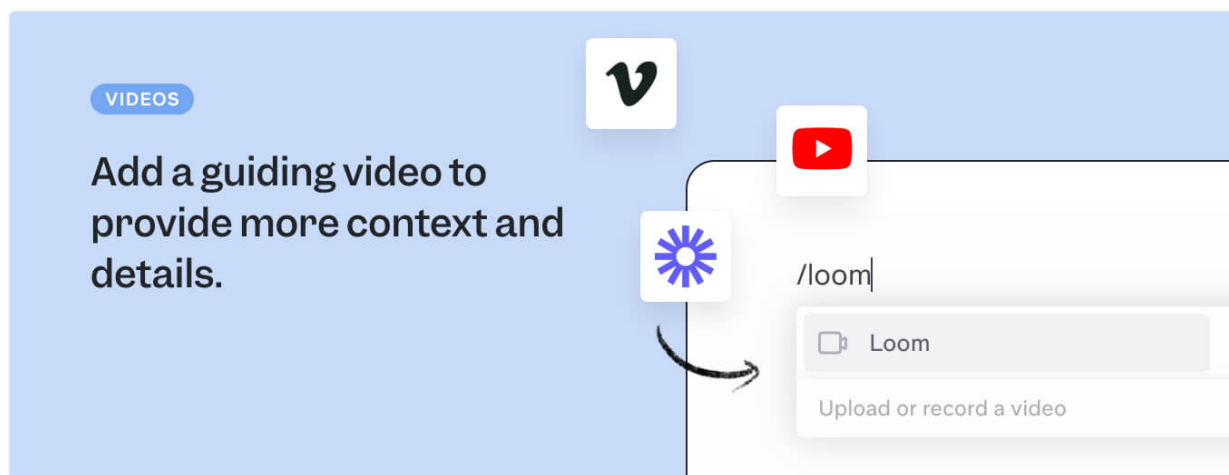
			    
	REGULAR_EXPRESSION	# ↓ MATCHES	MATCHES
	FAA*	7	FA,FA,FA,FA,FA,FA,FA
	FAA*	6	FA,FA,FA,FA,FA,FA
LLVDGRRIGV	FAA*	6	FA,FA,FA,FA,FA,FA
	FAA*	5	FA,FA,FA,FA,FA
	FAA*	5	FA,FA,FA,FA,FA
	FAA*	5	FA,FA,FA,FA,FA
	FAA*	5	FA,FA,FA,FA,FA
	FAA*	5	FA,FA,FA,FA,FA
	FAA*	5	FA,FA,FA,FA,FA
	FAA*	5	FA,FA,FA,FA,FA
	FAA*	4	FA,FA,FA,FA
Rows per page: 10 ▾ 1-10 of 995			<       >

Figure 11. Results Obtained 995 of 1000

## Tutorials

Have any videos or knowledge bases to link if new users need additional guidance?  
 Include them here.



## FAQs 🙋

Answer and document frequently asked questions below.

### Question

Answer

### Question

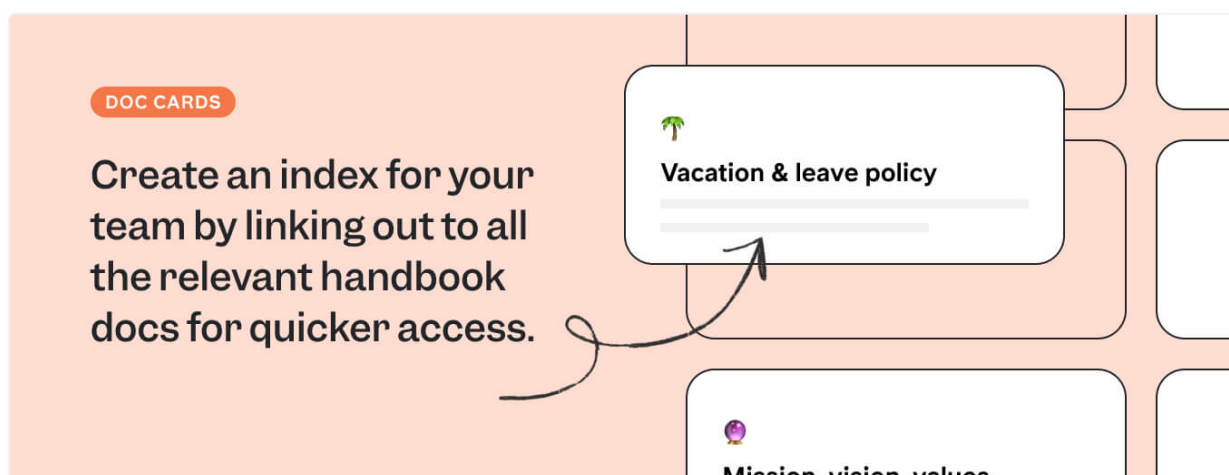
Answer

### Question

Answer



## Additional Resources 🧩

Include any additional information, forum or documentation that might be relevant to users here for easy access when questions arise.



## Support

This application was developed by Technotronic Engineer Victor Miguel Terron M. with advice from Ph.D. Miguel Angel Canseco Pérez. You can contact us:

-  [victor.terron@cimat.mx](mailto:victor.terron@cimat.mx)
  -  [miguel.canseco@uttlaxcala.edu.mx](mailto:miguel.canseco@uttlaxcala.edu.mx)
-



### **Make it your own**

Once edited to your liking, [save this template to your team's templates list](#) by clicking on the three dots on the right of the screen.