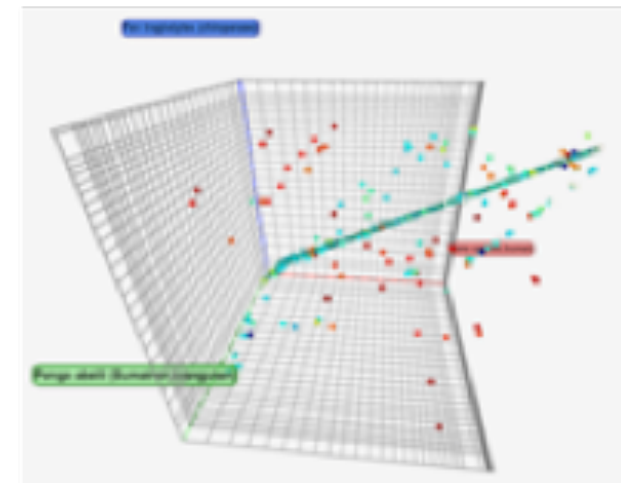
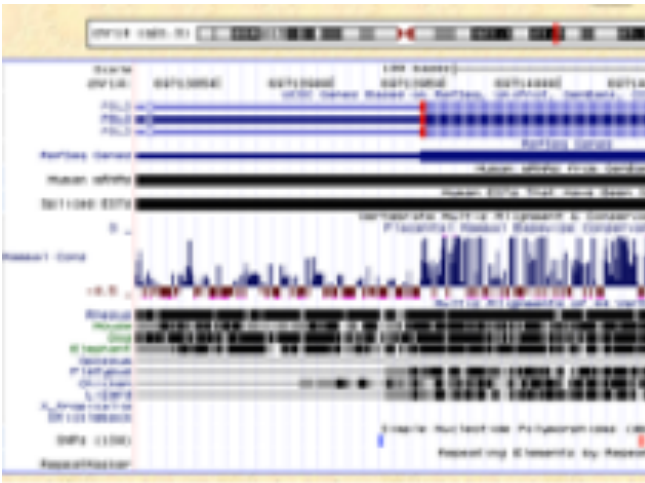


Computational Genomics

Introduction to Regular Expressions



Introduction to Regular Expressions

Go to regexr.com

Menu

Pattern Settings

My Patterns

Cheatsheet

RegExp Reference

Community Patterns

Help

RegExr is an online tool to **learn, build, & test** Regular Expressions (RegExp / RegEx).

- Supports **JavaScript & PHP/PCRE** RegExp.
- Results update in **real-time** as you type.
- Roll over** a match or expression for details.
- Validate patterns with suites of **Tests**.
- Save** & share expressions with others.
- Use **Tools** to explore your results.
- Full **RegExp Reference** with help & examples.
- Undo** & **Redo** with cmd-Z / Y in editors.
- Search for & rate **Community Patterns**.

Expression

<> JavaScript ▾

Flags ▾

/([A-Z])\w+/g

TextTestsNEW

29 matches (0.2ms)

RegExr was created by gskinner.com, and is proudly hosted by Media Temple.

Edit the Expression & Text to see matches. Roll over matches or the expression for details. PCRE & JavaScript flavors of RegEx are supported. Validate your expression with Tests mode.

The side bar includes a Cheatsheet, full Reference, and Help. You can also Save & Share with the Community, and view patterns you create or favorite in My Patterns.

Explore results with the Tools below. Replace & List output custom results. Details lists capture groups. Explain describes your expression in plain English.

Tools

ReplaceListDetailsExplain

Roll-over elements below to highlight in the Expression above. Click to open in Reference.

(Capturing group #1. Groups multiple tokens together and creates a capture group for extracting a substring or using a backreference.

[Character set. Match any character in the set.

A-Z Range. Matches a character in the range "A" to "Z" (char code 65 to 90). Case sensitive.

)

\w Word. Matches any word character (alphanumeric & underscore).

+ Quantifier. Match 1 or more of the preceding token.

Introduction to Regular Expressions

Select PCRE and Flags: global, multiline and extended

Menu

Pattern Settings

My Patterns

Cheatsheet

RegEx Reference

Community Patterns

Help

RegExr is an online tool to **learn, build, & test** Regular Expressions (RegEx / RegExp).

- Supports **JavaScript & PHP/PCRE** RegEx.
- Results update in **real-time** as you type.
- Roll over** a match or expression for details.
- Validate patterns with suites of **Tests**.
- Save** & share expressions with others.
- Use **Tools** to explore your results.
- Full **RegEx Reference** with help & examples.
- Undo** & **Redo** with cmd-Z / Y in editors.
- Search for & rate **Community Patterns**.

Expression

<> PCRE Flags

/([A-Z])\w+ /gmx

Text Tests NEW

RegExr was created by gskinner.com, and is proudly hosted by Media Temple.
Edit the Expression & Text to see matches. Roll over matches or the expression for details. PCRE & JavaScript RegEx are supported. Validate your expression with Tests mode.
The side bar includes a Cheatsheet, full Reference, and Help. You can also Save & Share with the Community, and create or favorite in My Patterns.
Explore results with the Tools below. Replace & List output custom results. Details lists capture groups. Explain describes your expression in plain English.

Tools

Replace List Details Explain

Roll-over elements below to highlight in the Expression above. Click to open in Reference.

(Capturing group #1. Groups multiple tokens together and creates a capture group for extracting a substring or using a backreference.

[Character set. Match any character in the set.

A-Z Range. Matches a character in the range "A" to "Z" (char code 65 to 90). Case sensitive.

\w Word. Matches any word character (alphanumeric & underscore).

+ Quantifier. Match 1 or more of the preceding token.

Introduction to Regular Expressions

A Regular Expression is a pattern describing a certain amount of text

- The () { } [] . * ? + ^ \$ are all special characters
- \ can be used to “escape” a special character, allowing that special character (i.e., () { } [] . * ? + ^ \$), to be searched for

Copy and Paste the following Text Extracted from this: [file](#). <- Link to text file

```
chrX 15561904 15562013 ENST00000252519.7_cds_0_0_chrX_15561905_r0-
chrX 15564023 15564218 ENST00000252519.7_cds_1_0_chrX_15564024_r0-
chrY 566252 566369 ENST00000252519.7_cds_2_0_chrX_15566253_r0-
chrX 15567725 15567826 ENST00000252519.7_cds_3_0_chrX_15567726_r0-
chrX 15570294 15570353 ENST00000252519.7_cds_4_0_chrX_15570295_r0-
chrX 15571623 15571796 ENST00000252519.7_cds_5_0_chrX_15571624_r0-
```

Expression

<> PCRE ▾

Flags ▾

/([A-Z])\w+/gmx

Text

Tests NEW

12 matches (0.3ms)

```
chrX→|15561904→|15562013→|ENST00000252519.7_cds_0_0_chrX_15561905_r0→|→|→|
chrX→|15564023→|15564218→|ENST00000252519.7_cds_1_0_chrX_15564024_r0→|→|→|
chrY→|566252→|→|→|566369→|→|→|ENST00000252519.7_cds_2_0_chrX_15566253_r0→|→|→|
chrX→|15567725→|15567826→|ENST00000252519.7_cds_3_0_chrX_15567726_r0→|→|→|
chrX→|15570294→|15570353→|ENST00000252519.7_cds_4_0_chrX_15570295_r0→|→|→|
chrX→|15571623→|15571796→|ENST00000252519.7_cds_5_0_chrX_15571624_r0→|→|→|
```

Introduction to Regular Expressions

- “\A” matches the beginning of a string (but not an internal line
- Regex: \AchrX

Expression

<> PCRE ▾ Flags ▾

/\AchrX/gmx

Text Tests NEW

1 match (0.1ms)

chrX→15561904→15562013→ENST00000252519.7_cds_0_0_chrX_15561905_r→0→|→
chrX→15564023→15564218→ENST00000252519.7_cds_1_0_chrX_15564024_r→0→|→
chrY→566252→|→566369→|→ENST00000252519.7_cds_2_0_chrX_15566253_r→0→|→
chrX→15567725→15567826→ENST00000252519.7_cds_3_0_chrX_15567726_r→0→|→
chrX→15570294→15570353→ENST00000252519.7_cds_4_0_chrX_15570295_r→0→|→
chrX→15571623→15571796→ENST00000252519.7_cds_5_0_chrX_15571624_r→0→|→

Tools

Replace List Details Explain

Roll-over elements below to highlight in the Expression above. Click to open in Reference.

\A Beginning of string. Matches the beginning of the string.

c Character. Matches a "c" character (char code 99). Case sensitive.

h Character. Matches a "h" character (char code 104). Case sensitive.

r Character. Matches a "r" character (char code 114). Case sensitive.

X Character. Matches a "X" character (char code 88). Case sensitive.

Introduction to Regular Expressions

- “\d” matches a digit class, same as [0–9]
- Regex: chr\d
- Regex: chr\d+
- Regex: \t\d+\t

Expression

<> PCRE ▾ 🚩 Flags ▾

/\t\d+\t/gmx

Text Tests **NEW**

13 matches (0.2ms)

chrX→15561904→15562013→ENST00000252519.7_cds_0_0_chrX_15561905_r→0→|→
chrX→15564023→15564218→ENST00000252519.7_cds_1_0_chrX_15564024_r→0→|→
chrY→566252→|→566369→|→ENST00000252519.7_cds_2_0_chrX_15566253_r→0→|→
chrX→15567725→15567826→ENST00000252519.7_cds_3_0_chrX_15567726_r→0→|→
chrX→15570294→15570353→ENST00000252519.7_cds_4_0_chrX_15570295_r→0→|→
chrX→15571623→15571796→ENST00000252519.7_cds_5_0_chrX_15571624_r→0→|→

Tools

Replace List Details Explain ✕

Roll-over elements below to highlight in the Expression above. Click to open in Reference.

\t Escaped character. Matches a TAB character (char code 9).

\d Digit. Matches any digit character (0-9).
+ Quantifier. Match 1 or more of the preceding token.

\t Escaped character. Matches a TAB character (char code 9).

Introduction to Regular Expressions

- “\D” matches a non-digit
- Regex: \D

Expression

<> PCRE ▾ Flags ▾

/\D/gmx

Text Tests NEW

181 matches (0.7ms)

chrX→15561904→15562013→ENST00000252519.7_cds_0_0_chrX_15561905_r→0→
chrX→15564023→15564218→ENST00000252519.7_cds_1_0_chrX_15564024_r→0→
chrY→566252→566369→ENST00000252519.7_cds_2_0_chrX_15566253_r→0→
chrX→15567725→15567826→ENST00000252519.7_cds_3_0_chrX_15567726_r→0→
chrX→15570294→15570353→ENST00000252519.7_cds_4_0_chrX_15570295_r→0→
chrX→15571623→15571796→ENST00000252519.7_cds_5_0_chrX_15571624_r→0→

Tools

Replace List Details Explain

Roll-over elements below to highlight in the Expression above. Click to open in Reference.

\D Not digit. Matches any character that is not a digit character (0-9).

Introduction to Regular Expressions

- “\s” matches a whitespace character
- Regex: \s
- Regex: \s+

Expression

<> PCRE ▾ 🚩 Flags ▾

/\s+/gmx

Text Tests **NEW**

35 matches (0.3ms)

chrX→15561904→15562013→ENST00000252519.7_cds_0_0_chrX_15561905_r→0→-
chrX→15564023→15564218→ENST00000252519.7_cds_1_0_chrX_15564024_r→0→-
chrY→566252→566369→ENST00000252519.7_cds_2_0_chrX_15566253_r→0→-
chrX→15567725→15567826→ENST00000252519.7_cds_3_0_chrX_15567726_r→0→-
chrX→15570294→15570353→ENST00000252519.7_cds_4_0_chrX_15570295_r→0→-
chrX→15571623→15571796→ENST00000252519.7_cds_5_0_chrX_15571624_r→0→-

Tools

Replace List Details Explain ✕

Roll-over elements below to highlight in the Expression above. Click to open in Reference.

\s

Whitespace. Matches any whitespace character (spaces, tabs, line breaks).

+

Quantifier. Match 1 or more of the preceding token.

Introduction to Regular Expressions

- “\S” matches anything BUT a whitespace
- Regex: \S

Expression

<> PCRE ▾ Flags ▾

/\S/gmx

Text Tests NEW

374 matches (1.4ms)

chrX→15561904→15562013→ENST00000252519.7_cds_0_0_chrX_15561905_r→0→
chrX→15564023→15564218→ENST00000252519.7_cds_1_0_chrX_15564024_r→0→
chrY→566252→566369→ENST00000252519.7_cds_2_0_chrX_15566253_r→0→
chrX→15567725→15567826→ENST00000252519.7_cds_3_0_chrX_15567726_r→0→
chrX→15570294→15570353→ENST00000252519.7_cds_4_0_chrX_15570295_r→0→
chrX→15571623→15571796→ENST00000252519.7_cds_5_0_chrX_15571624_r→0→

Tools

Replace List Details Explain

Roll-over elements below to highlight in the Expression above. Click to open in Reference.

\S Not whitespace. Matches any character that is not a whitespace character (spaces, tabs, line breaks).

Introduction to Regular Expressions

- “\t” matches a tab
- Regex: \t
- Regex: \t+

Expression

<> PCRE ▾ 🚩 Flags ▾

/\t+/gmx

Text Tests NEW

30 matches (0.2ms)

chrX→15561904→15562013→ENST00000252519.7_cds_0_0_chrX_15561905_r→0→-→
chrX→15564023→15564218→ENST00000252519.7_cds_1_0_chrX_15564024_r→0→-→
chrY→566252→→566369→→ENST00000252519.7_cds_2_0_chrX_15566253_r→0→-→
chrX→15567725→15567826→ENST00000252519.7_cds_3_0_chrX_15567726_r→0→-→
chrX→15570294→15570353→ENST00000252519.7_cds_4_0_chrX_15570295_r→0→-→
chrX→15571623→15571796→ENST00000252519.7_cds_5_0_chrX_15571624_r→0→-→

Tools

Replace List Details Explain ✕

Roll-over elements below to highlight in the Expression above. Click to open in Reference.

\t

Escaped character. Matches a TAB character (char code 9).

+

Quantifier. Match 1 or more of the preceding token.

Introduction to Regular Expressions

- “\w” matches an alphanumeric character
- Regex: \w
- Regex: \w+

Expression

<> PCRE ▾

Flags ▾

/\w+/gmx

TextTestsNEW

36 matches (0.4ms)

chrX→15561904>15562013>ENST00000252519.7_cds_0_0_chrX_15561905_r>0→| - ↵

chrX→15564023>15564218>ENST00000252519.7_cds_1_0_chrX_15564024_r>0→| - ↵

chrY→566252→|→|566369→|→|ENST00000252519.7_cds_2_0_chrX_15566253_r>0→| - ↵

chrX→15567725>15567826>ENST00000252519.7_cds_3_0_chrX_15567726_r>0→| - ↵

chrX→15570294>15570353>ENST00000252519.7_cds_4_0_chrX_15570295_r>0→| - ↵

chrX→15571623>15571796>ENST00000252519.7_cds_5_0_chrX_15571624_r>0→| - ↵

Tools

ReplaceListDetailsExplain✕

Roll-over elements below to highlight in the Expression above. Click to open in Reference.

?

\w Word. Matches any word character (alphanumeric & underscore).

+ Quantifier. Match 1 or more of the preceding token.

Introduction to Regular Expressions

- “\W” matches anything but an alphanumeric character
- Regex: \W

Expression <> PCRE Flags

`/\W/gmx`

Text **Tests** NEW 49 matches (0.3ms)

```
chrX→15561904→15562013→ENST00000252519.7_cds_0_0_chrX_15561905_r→0→|→
chrX→15564023→15564218→ENST00000252519.7_cds_1_0_chrX_15564024_r→0→|→
chrY→566252→→566369→→→ENST00000252519.7_cds_2_0_chrX_15566253_r→0→|→
chrX→15567725→15567826→ENST00000252519.7_cds_3_0_chrX_15567726_r→0→|→
chrX→15570294→15570353→ENST00000252519.7_cds_4_0_chrX_15570295_r→0→|→
chrX→15571623→15571796→ENST00000252519.7_cds_5_0_chrX_15571624_r→0→|→
```

Tools Replace List Details Explain ×


Roll-over elements below to highlight in the Expression above. Click to open in Reference.

`\W` **Not word.** Matches any character that is not a word character (alphanumeric & underscore).

Introduction to Regular Expressions

- “\Z” matches the end of a string (but not an internal line)
- Regex: `chrX\Z`

Expression

<> PCRE ▾  Flags ▾


/chrX\Z/gmx


Text Tests NEW

No match (0.2ms)

```
chrX→15561904→15562013→ENST00000252519.7_cds_0_0_chrX_15561905_r→0→|→  
chrX→15564023→15564218→ENST00000252519.7_cds_1_0_chrX_15564024_r→0→|→  
chrY→566252→|→|→566369→|→|→ENST00000252519.7_cds_2_0_chrX_15566253_r→0→|→  
chrX→15567725→15567826→ENST00000252519.7_cds_3_0_chrX_15567726_r→0→|→  
chrX→15570294→15570353→ENST00000252519.7_cds_4_0_chrX_15570295_r→0→|→  
chrX→15571623→15571796→ENST00000252519.7_cds_5_0_chrX_15571624_r→0→|→
```

Tools

Replace List Details Explain 

Roll-over elements below to highlight in the Expression above. Click to open in Reference. 

c Character. Matches a "c" character (char code 99). Case sensitive.

h Character. Matches a "h" character (char code 104). Case sensitive.

r Character. Matches a "r" character (char code 114). Case sensitive.

X Character. Matches a "X" character (char code 88). Case sensitive.

\Z End of string. Matches the end of the string.

Introduction to Regular Expressions

- “{ n or n, or n,m }” specifies an expected number of repetitions of the preceding pattern
- “{n}” The preceding item is matched exactly n times.
- “{n,}” The preceding item is matched n or more times.
- “{n,m}” The preceding item is matched at least n times but not more than m times.
- “[...]” creates a character class
 - Within the brackets, single characters can be placed
 - A dash (-) may be used to indicate a range such as a-z
- “.” Matches any single character except a newline
- “*” The preceding item will be matched zero or more times
- “?” The preceding item is optional and matched at most once
- “+” The preceding item will be matched one or more time
- “^” has two meaning:
 - matches the beginning of a line or string
 - indicates negation in a character class
- For example, “[^...]” matches every character except the ones inside brackets
 - “\$” matches the end of a line or string
 - “|” Separates alternate possibilities
 - “(..)” groups a particular pattern
- Regex: `^(chr.)\t([0-9]+)\t+([0-9]+)\t+ENST([0-9]+)`

Introduction to Regular Expressions

- **Regex:** `^(chr.)\t([0-9]+)\t+([0-9]+)\t+ENST([0-9]+)`

Expression

PCRE ▼

Flags ▼

```
/^ (chr.) \t ([0-9]+) \t+ ([0-9]+) \t+ ENST ([0-9]+) /gmx
```

Text

Tests

NEW

6 matches (0.2ms)

chrX→15561904→15562013→ENST00000252519.7_cds_0_0_chrX_15561905_r→|0→|→
chrX→15564023→15564218→ENST00000252519.7_cds_1_0_chrX_15564024_r→|0→|→
chrY→566252→|→566369→|→ENST00000252519.7_cds_2_0_chrX_15566253_r→|0→|→
chrX→15567725→15567826→ENST00000252519.7_cds_3_0_chrX_15567726_r→|0→|→
chrX→15570294→15570353→ENST00000252519.7_cds_4_0_chrX_15570295_r→|0→|→
chrX→15571623→15571796→ENST00000252519.7_cds_5_0_chrX_15571624_r→|0→|→

Introduction to Regular Expressions

- Copy and Paste the following text:

```
>ENST00000252519.7
YQSSLASWNYNTNITEENVQNMNNAGDKWSAFLKEQSTLAQMYEENVQNMNNAGPLQEIQNLTVKLQLQALQQNGSEEEENVQNMNN
AGSVLSEDKSKRLNTIL
```

Design a minimal expression to find the following sequences in one pattern:

ENVQNMNNAG and NMNNAGP

1.Align

Alignment01:

```
ENVQNMNNAG
....NMNNAGP
```

Alignment02:

```
ENVQNMNNAG
...NMNNAGP
```

Best Local Alignment:

```
ENVQNMNN.AG
....NMNNAGP
....NMNN.AG.
```

Regexp:

```
... (NMN+AG)
```


Introduction to Regular Expressions

Searching with Pattern 01: ENVQNMNNAG

Expression

<> PCRE ▾

🚩 Flags ▾

/ENVQNMNNAG/gmx

Text

Tests

NEW

2 matches (0.1ms)

>ENST000000252519.7
YQSSLASWNYNTNITEENVQNMNNAGDKWSAFLKEQSTLAQMYEENVQNMNNNAGPLQEIQNLTVKLQLQALQQNGSEEEENVQNMNNAGSVLSEDKSKRLNTIL

Introduction to Regular Expressions

Searching with Patter02: NMNNNAGP

Expression

<> PCRE ▾

🚩 Flags ▾

/NMNNNAGP/gmx

Text

Tests NEW

1 match (0.1ms)

>ENST000000252519.7
YQSSLASWNYNTNITEENVQNMNNAGDKWSAFLKEQSTLAQMYEENVQNMNNNAGPLQEIQNLTVKLQLQALQQNGSEEEENVQNMNNAGSVLSEDKSKRLNTIL

Introduction to Regular Expressions

Searching with Regexp: NMN+AG

Expression

<> PCRE ▾

🚩 Flags ▾

/NMN+AG/gmx

Text

Tests NEW

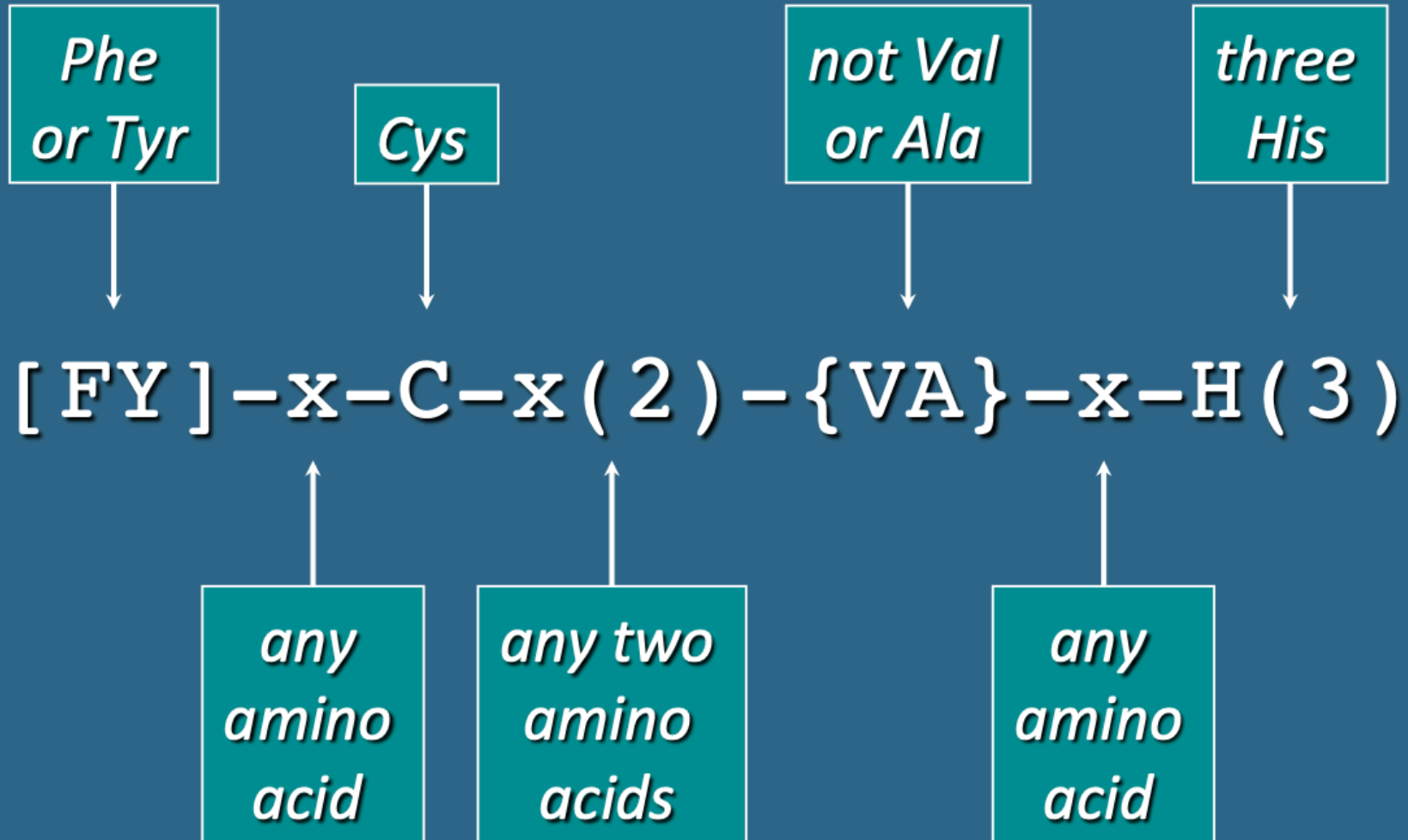
3 matches (0.1ms)

>ENST00000252519.7
YQSSLASWNYNTNITEENVQNMNNAGDKWSAFLKEQSTLAQMYEENVQNMNNAGPLQEIQNLTVKLQLQALQQNGSEEEENVQNMNNAGSVLSEDKSKRLNTIL

Introduction to Regular Expressions

- Patterns as formulas:

Patterns



Introduction to Regular Expressions

- HMM Logo Family: Piwi (PF02171)

HMM logo

HMM logos is one way of visualising profile HMMs. Logos provide a quick overview of the properties of an HMM in a graphical form. You can see a more detailed description of HMM logos and find out how you can interpret them [here](#).
[Less...](#)

If you find these logos useful in your own work, please consider citing the following article:

[HMM Logos for visualization of protein families](#): B. Schuster-Böckler, J. Schultz, S. Rahmann

BMC Bioinformatics (2004) 5:7

