사용자 매뉴얼

Release 0.0.32+6.g3c3d153.dirty

Changbum Hong

Dec 30, 2020

목차

1 Overview

최근 comprehensive genomic profiling을 사용한 대규모 코호트 연구들에 따르면 90%가 유용한 alteration을 가지고 있다고 보고되고 있다cite{Priestley:2019bp}. 엔젠바이오는 연구자들을 돕기 위해 323개의 암관련 유전자 (225 coding exon, 98 hotspot cover) 를 분석하는 ONCOaccuPanel을 제공한다. small nucleotide variants(SNVs), insertions/deletions(indels), copy-number variations(CNVs), splice variants, fusions과 함께 최근 다수의 genomic loci 분석을 기반으로 하는 tumor mutational burden(TMB) and microsatellite instability(MSI) 분석을 제공한다.

Variant type	Relevant exmaples
SNVs and indels	KRAS G12D, EGFR exon 19 deletions, BRAF V600E
Fusions	EGFR, ROS1, RET, ALK, NTRK1
Splice variants	MET exon 14
CNVs	HER2
MSI[1]	MSI-HIGH
TMB[2]	TMB-HIGH

Note: 로그인 창이 나타나지 않거나 로그인 버튼을 눌러도 연결되지 않는 경우에는 인터넷 또는 기관 내부망 (인트라넷) 연결이 활성화 되었는지 확인하거나, 아이디 및 비밀번호를 올바르게 입력했는지 확인한다.

Warning:

1. 선택한 암종에 따라 해석 (interpreation) 의 tier 정보가 달라진다.
2. 선택한 암종에 따라 white list가 달라진다.

1. NGS를 수행한 RUN 이름을 입력한다.
2. "Local Fastq Files"을 선택하여 로컬 컴퓨터의 FASTQ 파일 또는 "Server Fastq Files"를 선택1하여 원격지의 FASTQ 파일을 선택한다.
3. 분석할 FASTQ 파일을 선택한 후, 열기 버튼을 클릭한다.

This theme provides a responsive Material Design theme for Sphinx documentation. It derives heavily

[1] A numerical footnote.
[2] 모든 암종에 대해서 기준값 ()23 Muts/Mb) 을 제공하지만, TMB의 high/low는 암 종마다 기준값이 달라질 수 있음

from Material for MkDocs[3], and also uses code from Guzzle Sphinx Theme[4].

There are two methods to alter the theme. The first, and simplest, uses the options exposed through `html_theme_options` in `conf.py`. This site's options are:

```
html_theme_options = {
    'base_url': 'http://bashtage.github.io/sphinx-material/',
    'repo_url': 'https://github.com/bashtage/sphinx-material/',
    'repo_name': 'Material for Sphinx',
    'google_analytics_account': 'UA-XXXXX',
    'html_minify': True,
    'css_minify': True,
    'nav_title': 'Material Sphinx Demo',
    'logo_icon': '&#xe869',
    'globaltoc_depth': 2
}
```

The complete list of options with detailed explanations appears in `theme.conf`.

1.1 Configuration Options

`nav_title` Set the name to appear in the left sidebar/header. If not provided, uses html_short_title if defined, or html_title.

`touch_icon` Path to a touch icon, should be 152x152 or larger.

`google_analytics_account` Set to enable google analytics.

`repo_url` Set the repo url for the link to appear.

`repo_name` The name of the repo. If must be set if repo_url is set.

`repo_type` Must be one of github, gitlab or bitbucket.

`base_url` Specify a base_url used to generate sitemap.xml links. If not specified, then no sitemap will be built.

`globaltoc_depth` The maximum depth of the global TOC; set it to -1 to allow unlimited depth.

`globaltoc_collapse` If true, TOC entries that are not ancestors of the current page are collapsed.

`globaltoc_includehidden` If true, the global TOC tree will also contain hidden entries.

`theme_color` The theme color for mobile browsers. Hex Color without the leading #.

`color_primary` Primary color. Options are red, pink, purple, deep-purple, indigo, blue, light-blue, cyan, teal, green, light-green, lime, yellow, amber, orange, deep-orange, brown, grey, blue-grey, and white.

`color_accent` Accent color. Options are red, pink, purple, deep-purple, indigo, blue, light-blue, cyan, teal, green, light-green, lime, yellow, amber, orange, and deep-orange.

`html_minify` Minify pages after creation using htmlmin.

`html_prettify` Prettify pages, usually only for debugging.

`css_minify` Minify css files found in the output directory.

[3] https://squidfunk.github.io/mkdocs-material/
[4] https://github.com/guzzle/guzzle_sphinx_theme

`logo_icon` Set the logo icon. Should be a pre-escaped html string that indicates a unicode point, e.g., `'&#xe869'` which is used on this site.

`master_doc` Include the master document at the top of the page in the breadcrumb bar. You must also set this to true if you want to override the rootrellink block, in which case the content of the overridden block will appear

`nav_links` A list of dictionaries where each has three keys:

- `href`: The URL or pagename (str)
- `title`: The title to appear (str)
- `internal`: Flag indicating to use pathto to find the page. Set to False for external content. (bool)

`heroes` A `dict[str,str]` where the key is a pagename and the value is the text to display in the page's hero location.

`version_dropdown` A flag indicating whether the version drop down should be included. You must supply a JSON file to use this feature.

`version_dropdown_text` The text in the version dropdown button

`version_json` The location of the JSON file that contains the version information. The default assumes there is a file versions.json located in the root of the site.

`version_info` A dictionary used to populate the version dropdown. If this variable is provided, the static dropdown is used and any JavaScript information is ignored.

`table_classes` A list of classes to **not strip** from tables. All other classes are stripped, and the default table has no class attribute. Custom table classes need to provide the full style for the table.

1.2 Sidebars

You must set `html_sidebars` in order for the side bar to appear. There are four in the complete set.

```
html_sidebars = {
    "**": ["logo-text.html", "globaltoc.html", "localtoc.html", "searchbox.html"]
}
```

You can exclude any to hide a specific sidebar. For example, if this is changed to

```
html_sidebars = {
    "**": ["globaltoc.html"]
}
```

then only the global ToC would appear on all pages (** is a glob pattern).

1.3 Customizing the layout

You can customize the theme by overriding Jinja template blocks. For example, 'layout.html' contains several blocks that can be overridden or extended.

Place a 'layout.html' file in your project's '/_templates' directory.

```
mkdir source/_templates
touch source/_templates/layout.html
```

Then, configure your 'conf.py':

```
templates_path = ['_templates']
```

Finally, edit your override file `source/_templates/layout.html`:

```
{# Import the theme's layout. #}
{% extends '!layout.html' %}

{%- block extrahead %}
{# Add custom things to the head HTML tag #}
{# Call the parent block #}
{{ super() }}
{%- endblock %}
```

1.4 New Blocks

The theme has a small number of new blocks to simplify some types of customization:

`footerrel` Previous and next in the footer.

`font` The default font inline CSS and the class to the google API. Use this block when changing the font.

`fonticon` Block that contains the icon font. Use this to add additional icon fonts (e.g., FontAwesome[6]). You should probably call `{{ super() }}` at the end of the block to include the default icon font as well.

1.5 Version Dropdown

A version dropdown is available that lets you store multiple versions in a single site. The standard structure of the site, relative to the base is usually:

```
/
/devel
/v1.0.0
/v1.1.0
/v1.1.1
/v1.2.0
```

[6] https://fontawesome.com/

To use the version dropdown, you must set `version_dropdown` to `True` in the sites configuration.

There are two approaches, one which stores the version information in a JavaScript file and one which uses a dictionary in the configuration.

1.5.1 Using a Javascript File

The data used is read via javascript from a file. The basic structure of the file is a dictionary of the form [label, path].

This dictionary tells the dropdown that the release version is in the root of the site, the other versions are archived under their version number, and the development version is located in /devel.

Note: The advantage of this approach is that you can separate version information from the rendered documentation. This makes is easy to change the version dropdown in _older_ versions of the documentation to reflect additional versions that are released later. Changing the Javascript file changes the version dropdown content in all versions. This approach is used in statsmodels[7].

1.5.2 Using `conf.py`

Warning: This method has precedence over the JavaScript approach. If `version_info` is not empty in a site's `html_theme_options`, then the static approach is used.

The alternative uses a dictionary where the key is the title and the value is the target. The dictionary is part of the size configuration's `html_theme_options`.

The dictionary structure is nearly identical. Here you can use relative paths like in the JavaScript version. You can also use absolute paths.

Note: This approach is easier if you only want to have a fixed set of documentation, e.g., stable and devel.

---

[7] https://www.statsmodels.org/

2 Specimen

2.1 Body copy

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras arcu libero, mollis sed massa vel, ornare viverra ex. Mauris a ullamcorper lacus. Nullam urna elit, malesuada eget finibus ut, ullamcorper ac tortor. Vestibulum sodales pulvinar nisl, pharetra aliquet est. Quisque volutpat erat ac nisi accumsan tempor.

Sed suscipit, orci non pretium pretium, quam mi gravida metus, vel venenatis justo est condimentum diam. Maecenas non ornare justo. Nam a ipsum eros. Nulla aliquam orci sit amet nisl posuere malesuada. Proin aliquet nulla velit, quis ultricies orci feugiat et. `Ut tincidunt sollicitudin` tincidunt. Aenean ullamcorper sit amet nulla at interdum.

2.2 Headings

2.2.1 The 3rd level

The 4th level

The 5th level

The 6th level

2.3 Headings with secondary text

2.3.1 The 3rd level with secondary text

The 4th level with secondary text

The 5th level with secondary text

The 6th level with secondary text

2.4 Blockquotes

Morbi eget dapibus felis. Vivamus venenatis porttitor tortor sit amet rutrum. Pellentesque aliquet quam enim, eu volutpat urna rutrum a. Nam vehicula nunc mauris, a ultricies libero efficitur sed. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Sed molestie imperdiet consectetur.

2.4.1 Blockquote nesting

Sed aliquet, neque at rutrum mollis, neque nisi tincidunt nibh, vitae faucibus lacus nunc at lacus. Nunc scelerisque, quam id cursus sodales, lorem libero fermentum urna, ut efficitur elit ligula et nunc.

Mauris dictum mi lacus, sit amet pellentesque urna vehicula fringilla. Ut sit amet placerat ante. Proin sed elementum nulla. Nunc vitae sem odio. Suspendisse ac eros arcu. Vivamus orci erat, volutpat a tempor et, rutrum. eu odio.

> `Suspendisse rutrum facilisis risus`, eu posuere neque commodo a. Interdum et malesuada fames ac ante ipsum primis in faucibus. Sed nec leo bibendum, sodales mauris ut, tincidunt massa.

2.4.2 Other content blocks

Vestibulum vitae orci quis ante viverra ultricies ut eget turpis. Sed eu lectus dapibus, eleifend nulla varius, lobortis turpis. In ac hendrerit nisl, sit amet laoreet nibh.

> Praesent at `return target`, sodales nibh vel, tempor felis. Fusce vel lacinia lacus. Suspendisse rhoncus nunc non nisi iaculis ultrices. Donec consectetur mauris non neque imperdiet, eget volutpat libero.

```
var _extends = function(target) {
  for (var i = 1; i < arguments.length; i++) {
    var source = arguments[i];
    for (var key in source) {
      target[key] = source[key];
    }
  }
  return target;
};
```

2.5 Lists

2.5.1 Unordered lists

- Sed sagittis eleifend rutrum. Donec vitae suscipit est. Nullam tempus tellus non sem sollicitudin, quis rutrum leo facilisis. Nulla tempor lobortis orci, at elementum urna sodales vitae. In in vehicula nulla, quis ornare libero.
  - Duis mollis est eget nibh volutpat, fermentum aliquet dui mollis.
  - Nam vulputate tincidunt fringilla.

    - Nullam dignissim ultrices urna non auctor.

- Aliquam metus eros, pretium sed nulla venenatis, faucibus auctor ex. Proin ut eros sed sapien ullamcorper consequat. Nunc ligula ante, fringilla at aliquam ac, aliquet sed mauris.
- Nulla et rhoncus turpis. Mauris ultricies elementum leo. Duis efficitur accumsan nibh eu mattis. Vivamus tempus velit eros, porttitor placerat nibh lacinia sed. Aenean in finibus diam.

2.5.2 Ordered lists

1. Integer vehicula feugiat magna, a mollis tellus. Nam mollis ex ante, quis elementum eros tempor rutrum. Aenean efficitur lobortis lacinia. Nulla consectetur feugiat sodales.
2. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam ornare feugiat quam et egestas. Nunc id erat et quam pellentesque lacinia eu vel odio.
    1. Vivamus venenatis porttitor tortor sit amet rutrum. Pellentesque aliquet quam enim, eu volutpat urna rutrum a. Nam vehicula nunc mauris, a ultricies libero efficitur sed.
        1. Mauris dictum mi lacus
        2. Ut sit amet placerat ante
        3. Suspendisse ac eros arcu
    2. Morbi eget dapibus felis. Vivamus venenatis porttitor tortor sit amet rutrum. Pellentesque aliquet quam enim, eu volutpat urna rutrum a. Sed aliquet, neque at rutrum mollis, neque nisi tincidunt nibh.
    3. Pellentesque eget `var _extends` ornare tellus, ut gravida mi.

```
var _extends = function(target) {
  for (var i = 1; i < arguments.length; i++) {
    var source = arguments[i];
    for (var key in source) {
      target[key] = source[key];
    }
  }
  return target;
};
```

3. Vivamus id mi enim. Integer id turpis sapien. Ut condimentum lobortis sagittis. Aliquam purus tellus, faucibus eget urna at, iaculis venenatis nulla. Vivamus a pharetra leo.

2.5.3 Definition lists

Lorem ipsum dolor sit amet Sed sagittis eleifend rutrum. Donec vitae suscipit est. Nullam tempus tellus non sem sollicitudin, quis rutrum leo facilisis. Nulla tempor lobortis orci, at elementum urna sodales vitae. In in vehicula nulla.

Duis mollis est eget nibh volutpat, fermentum aliquet dui mollis. Nam vulputate tincidunt fringilla. Nullam dignissim ultrices urna non auctor.

Cras arcu libero Aliquam metus eros, pretium sed nulla venenatis, faucibus auctor ex. Proin ut eros sed sapien ullamcorper consequat. Nunc ligula ante, fringilla at aliquam ac, aliquet sed mauris.

2.6 Code blocks

2.6.1 Inline

Morbi eget `dapibus  felis`. Vivamus “venenatis porttitor“ tortor sit amet rutrum. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. `Pellentesque aliquet quam enim`, eu volutpat urna rutrum a.

Nam vehicula nunc `:::js return target` mauris, a ultricies libero efficitur sed. Sed molestie imperdiet consectetur. Vivamus a pharetra leo. Pellentesque eget ornare tellus, ut gravida mi. Fusce vel lacinia lacus.

2.6.2 Listing

```
var _extends = function(target) {
  for (var i = 1; i < arguments.length; i++) {
    var source = arguments[i];
    for (var key in source) {
      target[key] = source[key];
    }
  }
  return target;
};
```

2.7 Horizontal rules

Aenean in finibus diam. Duis mollis est eget nibh volutpat, fermentum aliquet dui mollis. Nam vulputate tincidunt fringilla. Nullam dignissim ultrices urna non auctor.

Integer vehicula feugiat magna, a mollis tellus. Nam mollis ex ante, quis elementum eros tempor rutrum. Aenean efficitur lobortis lacinia. Nulla consectetur feugiat sodales.

2.8 Data tables

Sollicitudo / Pellentesi	consectetur	adipiscing	elit	arcu	sed
Vivamus a pharetra	yes	yes	yes	yes	yes
Ornare viverra ex	yes	yes	yes	yes	yes
Mauris a ullamcorper	yes	yes	partial	yes	yes
Nullam urna elit	yes	yes	yes	yes	yes
Malesuada eget finibus	yes	yes	yes	yes	yes
Ullamcorper	yes	yes	yes	yes	yes
Vestibulum sodales	yes	•	yes	•	yes
Pulvinar nisl	yes	yes	yes	•	•
Pharetra aliquet est	yes	yes	yes	yes	yes
Sed suscipit	yes	yes	yes	yes	yes
Orci non pretium	yes	partial	•	•	•

Sed sagittis eleifend rutrum. Donec vitae suscipit est. Nullam tempus tellus non sem sollicitudin, quis rutrum leo facilisis. Nulla tempor lobortis orci, at elementum urna sodales vitae. In in vehicula nulla, quis ornare libero.

Left	Center	Right
Lorem	dolor	`amet`
ipsum	**sit**	

Vestibulum vitae orci quis ante viverra ultricies ut eget turpis. Sed eu lectus dapibus, eleifend nulla varius, lobortis turpis. In ac hendrerit nisl, sit amet laoreet nibh.

Table	with colgroups (Pandoc)
Lorem	ipsum dolor sit amet.
Sed sagittis	eleifend rutrum. Donec vitae suscipit est.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus nec ipsum a eros convallis facilisis eget at leo. Cras eu pulvinar eros, at accumsan dolor. Ut gravida massa sed eros imperdiet fermentum. Donec ac diam ut lorem consequat laoreet. Maecenas at ex diam. Phasellus tincidunt orci felis, nec commodo nisl aliquet ac. Aenean eget ornare tellus. Nullam vel nunc quis nisi sodales finibus in ut metus. Praesent ultrices mollis leo, auctor volutpat eros consectetur in. Sed ac odio nisi. Cras aliquet ultrices nisl ac mattis. Nulla a dui velit. Proin et ipsum quis metus auctor viverra. Proin suscipit massa quis magna mattis, vel tincidunt quam tincidunt. Vestibulum nec feugiat metus, nec scelerisque eros. Ut ultricies ornare aliquam.

2.9 Section II

Proin ac mi tempor, ullamcorper ante at, sodales augue. Duis enim turpis, volutpat eget consectetur id, facilisis vel nisl. Sed non leo aliquam, tempus nisl eu, vestibulum enim. Suspendisse et leo imperdiet, pulvinar lacus sed, commodo felis.

Note: Praesent elit mi, pretium nec pellentesque eget, ultricies euismod turpis.

2.9.1 Sub section

In lobortis elementum tempus. Nam facilisis orci neque, eget vestibulum lectus imperdiet sed. Aenean ac eros sollicitudin, accumsan turpis ac, faucibus arcu.

2.10 Section III

Donec sodales, velit ac sagittis fermentum, metus ante pharetra ex, ac eleifend erat ligula in lacus. Donec tincidunt urna est, non mollis turpis lacinia sit amet. Duis ac facilisis libero, ut interdum nibh. Sed rutrum dapibus pharetra. Ut ac luctus nisi, vitae volutpat arcu. Vivamus venenatis eu nibh ut consectetur. Cras tincidunt dui nisi, et facilisis eros feugiat nec.

Fusce ante:

- libero
- consequat quis facilisis id
- sollicitudin et nisl.

Aliquam diam mi, vulputate nec posuere id, consequat id elit. Ut feugiat lectus quam, sed aliquet augue placerat nec. Sed volutpat leo ac condimentum ullamcorper. Vivamus eleifend magna tellus, sit amet porta nunc ultrices eget. Nullam id laoreet ex. Nam ultricies, ante et molestie mollis, magna sem porta libero, sed molestie neque risus ut purus. Ut tellus sapien, auctor a lacus eu, iaculis congue ex.

Duis et nisi a odio **scelerisque** sodales ac ut sapien. Ut eleifend blandit velit luctus euismod. Curabitur at pulvinar mi. Cras molestie lorem non accumsan gravida. Sed vulputate, ligula ut tincidunt congue, metus risus luctus lacus, sed rhoncus ligula turpis non erat. Phasellus est est, sollicitudin ut elementum vel, placerat in orci. Proin molestie posuere dolor sit amet convallis. Donec id urna vel lacus ultrices pulvinar sit amet id metus. Donec in venenatis ante. Nam eu rhoncus leo. Quisque posuere, leo vel porttitor malesuada, nisi urna dignissim justo, vel consectetur purus elit in mauris. Vestibulum lectus arcu, varius ut ligula quis, viverra gravida sem.

Warning: Pellentesque in enim leo.

2.11 Images

2.11.1 Default Alignment

2.11.2 Center Alignment

2.11.3 Right Alignment

3 Jupyter Notebooks

The nbsphinx extension[8] allow notebooks to be seemlessly integrated into a Sphinx website. This page demonstrates how notebooks are rendered.

3.1 facebook

asfasdfasdfasdf	asdfasdf	asdfasdf	asdfasdfasdf	asdfasdfasdfasdf
1324	2	2	1	1
1234	2	2	1	1
1234	2	2	2	1

Stretch/Untouched	ProbDistribution	Accuracy
Stretched	Gaussian	.843

[1]:
```
from IPython.display import HTML, display
import tabulate
table = [["Sun",696000,1989100000],
         ["Earth",6371,5973.6],
         ["Moon",1737,73.5],
         ["Mars",3390,641.85]]
display(HTML(tabulate.tabulate(table, tablefmt='html')))
```

[8] https://github.com/spatialaudio/nbsphinx

```
<IPython.core.display.HTML object>
```

3.2 DataFrames

pandas DataFrames are rendered with useful markup.

[2]:
```
import numpy as np
import pandas as pd

df = pd.DataFrame({'ints': [1, 2, 3],
                   'floats': [np.pi, np.exp(1), (1+np.sqrt(5))/2],
                   'strings': ['aardvark', 'bananarama', 'charcuterie' ]})

df
```

[2]:

	ints	floats	strings
0	1	3.141593	aardvark
1	2	2.718282	bananarama
2	3	1.618034	charcuterie

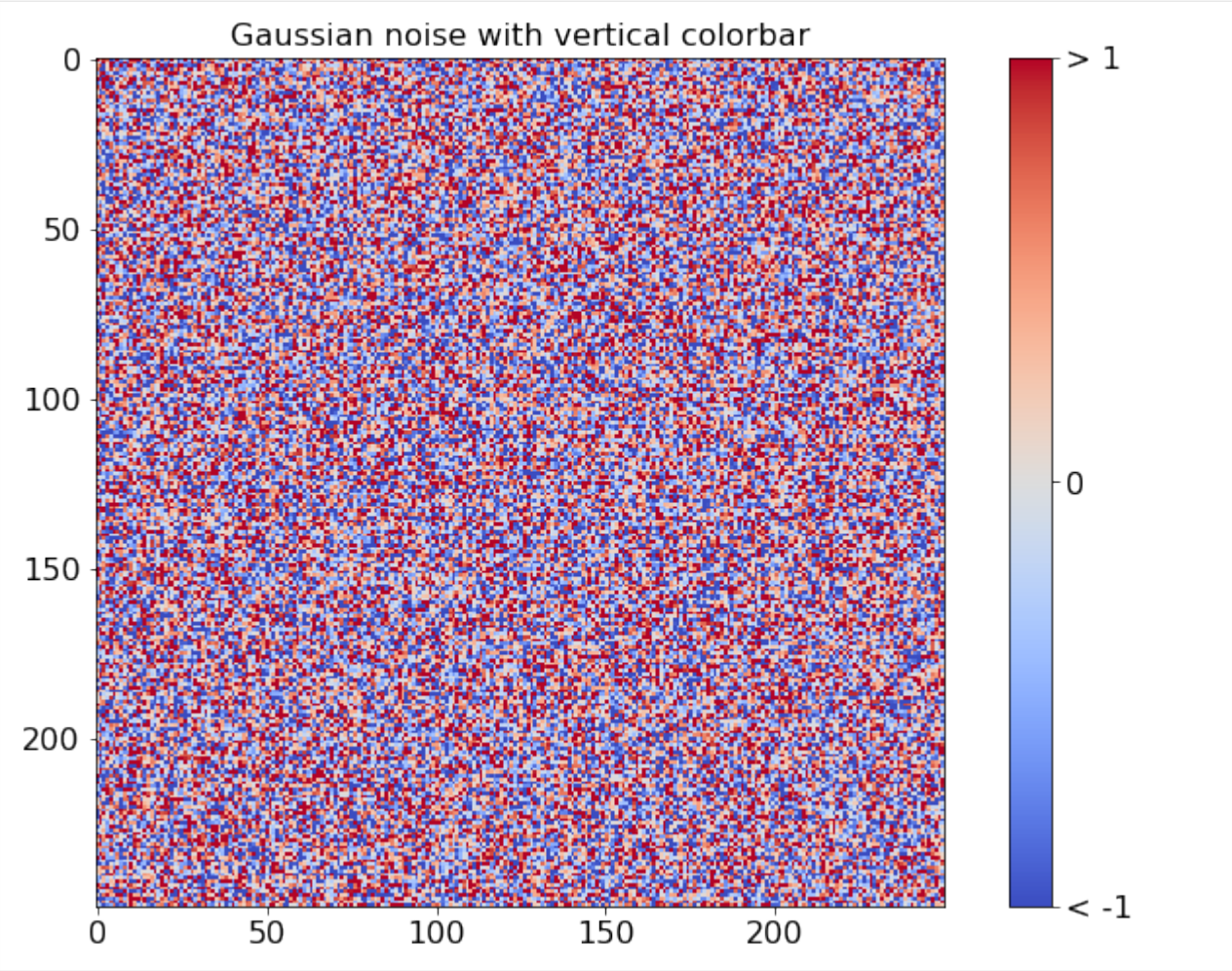
3.3 Plots and Figures

matplotlib can be used to produce plots in notebooks

This example comes from the matplotlib gallery[9].

[3]:
```
%matplotlib inline

import numpy as np
import matplotlib.pyplot as plt
from matplotlib import cm

fig, ax = plt.subplots(figsize=(12,8))

data = np.clip(np.random.randn(250, 250), -1, 1)

cax = ax.imshow(data, interpolation='nearest', cmap=cm.coolwarm)
ax.set_title('Gaussian noise with vertical colorbar', fontsize=16)
plt.tick_params(labelsize=16)

# Add colorbar, make sure to specify tick locations to match desired ticklabels
cbar = fig.colorbar(cax, ticks=[-1, 0, 1])
cbar.ax.set_yticklabels(['< -1', '0', '> 1'])  # vertically oriented colorbar
cbar.ax.tick_params(labelsize=16)
```

[9] https://matplotlib.org/3.1.1/gallery/ticks_and_spines/colorbar_tick_labelling_demo.html#sphx-glr-gallery-ticks-and-spines-colorbar-tick-la


Gaussian noise with vertical colorbar
0
50
100
150
200
0
50
100
150
200
> 1
0
< -1

4 NumPy Docstrings

This page shows how `autosummary` works with `numpydoc` and a NumPy-style docstring.

`Polynomial`(coef[, domain, window])	A power series class.

4.1 numpy.polynomial.Polynomial

`class numpy.polynomial.Polynomial(coef, domain=None, window=None)`

A power series class.

The Polynomial class provides the standard Python numerical methods '+', '-', '*', '//', '%', 'divmod', '**', and '()' as well as the attributes and methods listed in the ABCPolyBase documentation.

Parameters

coef [array_like] Polynomial coefficients in order of increasing degree, i.e., `(1, 2, 3)` give `1 + 2*x + 3*x**2`.

domain [(2,) array_like, optional] Domain to use. The interval `[domain[0], domain[1]]` is mapped to the interval `[window[0], window[1]]` by shifting and scaling. The default value is `[-1, 1]`.

window [(2,) array_like, optional] Window, see domain for its use. The default value is `[-1, 1]`.

New in version 1.6.0.

Methods

`__call__`(arg)	Call self as a function.
`basis`(deg[, domain, window])	Series basis polynomial of degree deg.
`cast`(series[, domain, window])	Convert series to series of this class.
`convert`([domain, kind, window])	Convert series to a different kind and/or domain and/or window.
`copy`()	Return a copy.
`cutdeg`(deg)	Truncate series to the given degree.
`degree`()	The degree of the series.
`deriv`([m])	Differentiate.
`fit`(x, y, deg[, domain, rcond, full, w, window])	Least squares fit to data.
`fromroots`(roots[, domain, window])	Return series instance that has the specified roots.
`has_samecoef`(other)	Check if coefficients match.
`has_samedomain`(other)	Check if domains match.
`has_sametype`(other)	Check if types match.
`has_samewindow`(other)	Check if windows match.
`identity`([domain, window])	Identity function.
`integ`([m, k, lbnd])	Integrate.
`linspace`([n, domain])	Return x, y values at equally spaced points in domain.
`mapparms`()	Return the mapping parameters.
`roots`()	Return the roots of the series polynomial.
`trim`([tol])	Remove trailing coefficients
`truncate`(size)	Truncate series to length size.

Methods

`basis`(deg[, domain, window])	Series basis polynomial of degree deg.
`cast`(series[, domain, window])	Convert series to series of this class.
`convert`([domain, kind, window])	Convert series to a different kind and/or domain and/or window.
`copy`()	Return a copy.
`cutdeg`(deg)	Truncate series to the given degree.
`degree`()	The degree of the series.
`deriv`([m])	Differentiate.
`fit`(x, y, deg[, domain, rcond, full, w, window])	Least squares fit to data.
`fromroots`(roots[, domain, window])	Return series instance that has the specified roots.
`has_samecoef`(other)	Check if coefficients match.
`has_samedomain`(other)	Check if domains match.
`has_sametype`(other)	Check if types match.
`has_samewindow`(other)	Check if windows match.
`identity`([domain, window])	Identity function.
`integ`([m, k, lbnd])	Integrate.

continues on next page

Table 3 – continued from previous page

linspace([n, domain])	Return x, y values at equally spaced points in domain.
mapparms()	Return the mapping parameters.
roots()	Return the roots of the series polynomial.
trim([tol])	Remove trailing coefficients
truncate(size)	Truncate series to length size.

4.1.1 numpy.polynomial.Polynomial.basis

method

classmethod Polynomial.basis(deg, domain=None, window=None)

Series basis polynomial of degree deg.

Returns the series representing the basis polynomial of degree deg.

New in version 1.7.0.

Parameters

deg [int] Degree of the basis polynomial for the series. Must be >= 0.

domain [{None, array_like}, optional] If given, the array must be of the form [beg, end], where beg and end are the endpoints of the domain. If None is given then the class domain is used. The default is None.

window [{None, array_like}, optional] If given, the resulting array must be if the form [beg, end], where beg and end are the endpoints of the window. If None is given then the class window is used. The default is None.

Returns

new_series [series] A series with the coefficient of the deg term set to one and all others zero.

4.1.2 numpy.polynomial.Polynomial.cast

method

classmethod Polynomial.cast(series, domain=None, window=None)

Convert series to series of this class.

The series is expected to be an instance of some polynomial series of one of the types supported by by the numpy.polynomial module, but could be some other class that supports the convert method.

New in version 1.7.0.

Parameters

series [series] The series instance to be converted.

domain [{None, array_like}, optional] If given, the array must be of the form [beg, end], where beg and end are the endpoints of the domain. If None is given then the class domain is used. The default is None.

window [{None, array_like}, optional] If given, the resulting array must be if the form [beg, end], where beg and end are the endpoints of the window. If None is given then the class window is used. The default is None.

Returns

new_series [series] A series of the same kind as the calling class and equal to series when evaluated.

See also:

`convert` similar instance method

4.1.3 numpy.polynomial.Polynomial.convert

method

`Polynomial.convert(domain=None, kind=None, window=None)`
Convert series to a different kind and/or domain and/or window.

Parameters

domain [array_like, optional] The domain of the converted series. If the value is None, the default domain of kind is used.

kind [class, optional] The polynomial series type class to which the current instance should be converted. If kind is None, then the class of the current instance is used.

window [array_like, optional] The window of the converted series. If the value is None, the default window of kind is used.

Returns

new_series [series] The returned class can be of different type than the current instance and/or have a different domain and/or different window.

Notes

Conversion between domains and class types can result in numerically ill defined series.

4.1.4 numpy.polynomial.Polynomial.copy

method

`Polynomial.copy()`
Return a copy.

Returns

new_series [series] Copy of self.

4.1.5 numpy.polynomial.Polynomial.cutdeg

method

`Polynomial.cutdeg(deg)`
Truncate series to the given degree.

Reduce the degree of the series to deg by discarding the high order terms. If deg is greater than the current degree a copy of the current series is returned. This can be useful in least squares where the coefficients of the high degree terms may be very small.

New in version 1.5.0.

Parameters

deg [non-negative int] The series is reduced to degree deg by discarding the high order terms. The value of deg must be a non-negative integer.

Returns

new_series [series] New instance of series with reduced degree.

4.1.6 numpy.polynomial.Polynomial.degree

method

`Polynomial.degree()`
The degree of the series.

New in version 1.5.0.

Returns

degree [int] Degree of the series, one less than the number of coefficients.

4.1.7 numpy.polynomial.Polynomial.deriv

method

`Polynomial.deriv(m=1)`
Differentiate.

Return a series instance of that is the derivative of the current series.

Parameters

m [non-negative int] Find the derivative of order m.

Returns

new_series [series] A new series representing the derivative. The domain is the same as the domain of the differentiated series.

4.1.8 numpy.polynomial.Polynomial.fit

method

classmethod Polynomial.fit(x, y, deg, domain=None, rcond=None, full=False, w=None, window=None)

Least squares fit to data.

Return a series instance that is the least squares fit to the data y sampled at x. The domain of the returned instance can be specified and this will often result in a superior fit with less chance of ill conditioning.

Parameters

x [array_like, shape (M,)] x-coordinates of the M sample points (x[i], y[i]).

y [array_like, shape (M,) or (M, K)] y-coordinates of the sample points. Several data sets of sample points sharing the same x-coordinates can be fitted at once by passing in a 2D-array that contains one dataset per column.

deg [int or 1-D array_like] Degree(s) of the fitting polynomials. If deg is a single integer all terms up to and including the deg'th term are included in the fit. For NumPy versions >= 1.11.0 a list of integers specifying the degrees of the terms to include may be used instead.

domain [{None, [beg, end], []}, optional] Domain to use for the returned series. If None, then a minimal domain that covers the points x is chosen. If [] the class domain is used. The default value was the class domain in NumPy 1.4 and None in later versions. The [] option was added in numpy 1.5.0.

rcond [float, optional] Relative condition number of the fit. Singular values smaller than this relative to the largest singular value will be ignored. The default value is len(x)*eps, where eps is the relative precision of the float type, about 2e-16 in most cases.

full [bool, optional] Switch determining nature of return value. When it is False (the default) just the coefficients are returned, when True diagnostic information from the singular value decomposition is also returned.

w [array_like, shape (M,), optional] Weights. If not None the contribution of each point (x[i],y[i]) to the fit is weighted by w[i]. Ideally the weights are chosen so that the errors of the products w[i]*y[i] all have the same variance. The default value is None.

New in version 1.5.0.

window [{[beg, end]}, optional] Window to use for the returned series. The default value is the default class domain

New in version 1.6.0.

Returns

new_series [series] A series that represents the least squares fit to the data and has the domain specified in the call.

[resid, rank, sv, rcond] [list] These values are only returned if full = True

resid - sum of squared residuals of the least squares fit rank - the numerical rank of the scaled Vandermonde matrix sv - singular values of the scaled Vandermonde matrix rcond - value of rcond.

For more details, see linalg.lstsq.

4.1.9 numpy.polynomial.Polynomial.fromroots

method

`classmethod Polynomial.fromroots(roots, domain=[], window=None)`
Return series instance that has the specified roots.

Returns a series representing the product `(x - r[0])*(x - r[1])*...*(x - r[n-1])`, where `r` is a list of roots.

Parameters

roots [array_like] List of roots.

domain [{[], None, array_like}, optional] Domain for the resulting series. If None the domain is the interval from the smallest root to the largest. If [] the domain is the class domain. The default is [].

window [{None, array_like}, optional] Window for the returned series. If None the class window is used. The default is None.

Returns

new_series [series] Series with the specified roots.

4.1.10 numpy.polynomial.Polynomial.has_samecoef

method

`Polynomial.has_samecoef(other)`
Check if coefficients match.

New in version 1.6.0.

Parameters

other [class instance] The other class must have the `coef` attribute.

Returns

bool [boolean] True if the coefficients are the same, False otherwise.

4.1.11 numpy.polynomial.Polynomial.has_samedomain

method

`Polynomial.has_samedomain(other)`
Check if domains match.

New in version 1.6.0.

Parameters

other [class instance] The other class must have the `domain` attribute.

Returns

bool [boolean] True if the domains are the same, False otherwise.

4.1.12 numpy.polynomial.Polynomial.has_sametype

method

`Polynomial.has_sametype(other)`
Check if types match.

New in version 1.7.0.

Parameters

other [object] Class instance.

Returns

bool [boolean] True if other is same class as self

4.1.13 numpy.polynomial.Polynomial.has_samewindow

method

`Polynomial.has_samewindow(other)`
Check if windows match.

New in version 1.6.0.

Parameters

other [class instance] The other class must have the `window` attribute.

Returns

bool [boolean] True if the windows are the same, False otherwise.

4.1.14 numpy.polynomial.Polynomial.identity

method

`classmethod Polynomial.identity(domain=None, window=None)`
Identity function.

If p is the returned series, then `p(x) == x` for all values of x.

Parameters

domain [{None, array_like}, optional] If given, the array must be of the form `[beg, end]`, where beg and end are the endpoints of the domain. If None is given then the class domain is used. The default is None.

window [{None, array_like}, optional] If given, the resulting array must be if the form `[beg, end]`, where beg and end are the endpoints of the window. If None is given then the class window is used. The default is None.

Returns

new_series [series] Series of representing the identity.

4.1.15 numpy.polynomial.Polynomial.integ

method

Polynomial.integ(m=1, k=[], lbnd=None)
Integrate.

Return a series instance that is the definite integral of the current series.

Parameters

m [non-negative int] The number of integrations to perform.

k [array_like] Integration constants. The first constant is applied to the first integration, the second to the second, and so on. The list of values must less than or equal to m in length and any missing values are set to zero.

lbnd [Scalar] The lower bound of the definite integral.

Returns

new_series [series] A new series representing the integral. The domain is the same as the domain of the integrated series.

4.1.16 numpy.polynomial.Polynomial.linspace

method

Polynomial.linspace(n=100, domain=None)
Return x, y values at equally spaced points in domain.

Returns the x, y values at n linearly spaced points across the domain. Here y is the value of the polynomial at the points x. By default the domain is the same as that of the series instance. This method is intended mostly as a plotting aid.

New in version 1.5.0.

Parameters

n [int, optional] Number of point pairs to return. The default value is 100.

domain [{None, array_like}, optional] If not None, the specified domain is used instead of that of the calling instance. It should be of the form [beg,end]. The default is None which case the class domain is used.

Returns

x, y [ndarray] x is equal to linspace(self.domain[0], self.domain[1], n) and y is the series evaluated at element of x.

4.1.17 numpy.polynomial.Polynomial.mapparms

method

Polynomial.mapparms()
Return the mapping parameters.

The returned values define a linear map off + scl*x that is applied to the input arguments before the series is evaluated. The map depends on the domain and window; if the current

domain is equal to the window the resulting map is the identity. If the coefficients of the series instance are to be used by themselves outside this class, then the linear function must be substituted for the x in the standard representation of the base polynomials.

Returns

off, scl [float or complex] The mapping function is defined by `off + scl*x`.

Notes

If the current domain is the interval `[l1, r1]` and the window is `[l2, r2]`, then the linear mapping function L is defined by the equations:

```
L(l1) = l2
L(r1) = r2
```

4.1.18 numpy.polynomial.Polynomial.roots

method

`Polynomial.roots()`

Return the roots of the series polynomial.

Compute the roots for the series. Note that the accuracy of the roots decrease the further outside the domain they lie.

Returns

roots [ndarray] Array containing the roots of the series.

4.1.19 numpy.polynomial.Polynomial.trim

method

`Polynomial.trim(tol=0)`

Remove trailing coefficients

Remove trailing coefficients until a coefficient is reached whose absolute value greater than tol or the beginning of the series is reached. If all the coefficients would be removed the series is set to `[0]`. A new series instance is returned with the new coefficients. The current instance remains unchanged.

Parameters

tol [non-negative number.] All trailing coefficients less than tol will be removed.

Returns

new_series [series] Contains the new set of coefficients.

4.1.20 numpy.polynomial.Polynomial.truncate

method

Polynomial.truncate(size)

Truncate series to length size.

Reduce the series to length size by discarding the high degree terms. The value of size must be a positive integer. This can be useful in least squares where the coefficients of the high degree terms may be very small.

Parameters

size [positive int] The series is reduced to length size by discarding the high degree terms. The value of size must be a positive integer.

Returns

new_series [series] New instance of series with truncated coefficients.

Properties

domain
maxpower
nickname
window

4.1.21 numpy.polynomial.Polynomial.domain

attribute

Polynomial.domain = array([-1, 1])

4.1.22 numpy.polynomial.Polynomial.maxpower

attribute

Polynomial.maxpower = 100

4.1.23 numpy.polynomial.Polynomial.nickname

attribute

Polynomial.nickname = 'poly'

4.1.24 numpy.polynomial.Polynomial.window

attribute

```
Polynomial.window = array([-1, 1])
```

5 Microsoft 365 docs navigation guide

This topic provides some tips and tricks for navigating the Microsoft 365 technical documentation space.

5.1 Impact to customers who don't transition

The following table summarizes the impact to customers who don't transition from a Microsoft 365 Business Preview subscription to a Microsoft 365 Business subscription.

5.2 Hub page

The Microsoft 365 hub page can be found at https://aka.ms/microsoft365docs and is the entry point for finding relevant Microsoft 365 content.

You can always navigate back to this page by selecting **Microsoft 365** from the header at the top of every page within the Microsoft 365 technical documentation set:

6

JupyterLab Demo Start

JupyterLab: The next generation user interface for Project Jupyter

https://github.com/jupyter/jupyterlab

It started as a collaboration between:

- Project Jupyter
- Bloomberg
- (then) Continuum

6.1 Tables

and now involves many other1 people from many other places (not purely academic or business)

1: 주석

```
function fancyAlert(arg) {
  if(arg) {
    $.facebox({div:'#foo'})
  }
}
```

6.2 1) Building blocks of interactive computing

6.2.1 Start with the launcher

Use it to open different activities:

- Notebook
- Console
- Editor
- Terminal

6.2.2 Notebooks

- Open example notebooks to show that notebooks still work
- Collapse input/output
- Drag and drop cells

6.2.3 Demonstrate left panel plugins:

- File Browser (file operations, context menu, including drag and drop)
- Running
- Command Palette (fuzzy searching for 'new')

6.2.4 Markdown example

- Open `markdown_python.md` in the File Editor
- View the rendered markdown, arrange side by side
- Attach a Kernel/Console and run the code by selecting blocks and pressing `Shift+Enter`

6.2.5 Arrange the building blocks in the main area

The dock panel allows you to arrange the activites into an arbitrary layout.
Tabs and single document mode allow you to focus.

6.3 4) File handlers

JupyterLab has a powerful and extensible architecture for handling a wide range of file formats:

- CSV
  - `./data/iris.csv` (small)
  - `TCGA_Data` (small to medium)
  - Urban_Data_Challenge: `data/big.csv`
- Images
  - `data/hubble.png`
- Vega-Lite
  - `data/vega.vl.json`
- Open DC museum GeoJSON file from OpenData DC[10]: `data/Museums_in_DC.geojson`
- Notebook demonstrating bqplot widgets: `notebooks/bqplot.ipynb`

[10] http://opendata.dc.gov/datasets/2e65fc16edc3481989d2cc17e6f8c533_54

6.4 5) Find and Replace

first class support for find and replace across JupyterLab, currently supported in notebooks and text files and is extensible for other widgets who wish to support it.

6.5 6) Status Bar

We have integrated the JupyterLab Status Bar package package into the core distribution. Extensions can add their own status to it as well

6.6 7) Printing

A printing system allows extensions to customize how documents and activities are printed.

6.7 8) JupyterHub

We now include the JupyterHub extension as a core JupyterLab extension, so you no longer need to install @jupyterlab/hub-extension (supporting multi-user + authentication workflows)

6.8 9) Plugin architecture

The genius of open-source is being able to shape your tools to your heart's content.

Just like Jupyter is built on top of building blocks of the protocol and message spec, you can build on this platform for your workflow.

- Everything in JupyterLab is an extension, including everything we have demoed
- Extensions are just `npm` packages with metadata
- Anyone can create, package, ship plugins
- Extension can, for example:
  - Add things to command palette, menu
  - Add viewers for documents
  - Expose other controls (e.g., manage a spark cluster?)
  - Provide more capabilities to the system

6.9 What will you build?

7 Markdown

Sphinx can be configured to use markdown using the recommonmark[11] extension. recommonmark is strict and does not natively support tables or common extensions to markdown.

7.1 Body copy

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras arcu libero, mollis sed massa vel, ornare viverra ex. Mauris a ullamcorper lacus. Nullam urna elit, malesuada eget finibus ut, ullamcorper ac tortor. Vestibulum sodales pulvinar nisl, pharetra aliquet est. Quisque volutpat erat ac nisi accumsan tempor.

Sed suscipit, orci non pretium pretium, quam mi gravida metus, vel venenatis justo est condimentum diam. Maecenas non ornare justo. Nam a ipsum eros. Nulla aliquam orci sit amet nisl posuere malesuada. Proin aliquet nulla velit, quis ultricies orci feugiat et. `Ut tincidunt sollicitudin` tincidunt. Aenean ullamcorper sit amet nulla at interdum.

7.2 Headings

7.2.1 The 3rd level

The 4th level

The 5th level

The 6th level

7.3 Headings with secondary text

7.3.1 The 3rd level with secondary text

[11] https://github.com/readthedocs/recommonmark

The 4th level with secondary text

The 5th level with secondary text

The 6th level with secondary text

7.4 Blockquotes

> Morbi eget dapibus felis. Vivamus venenatis porttitor tortor sit amet rutrum. Pellentesque aliquet quam enim, eu volutpat urna rutrum a. Nam vehicula nunc mauris, a ultricies libero efficitur sed. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Sed molestie imperdiet consectetur.

7.4.1 Blockquote nesting

> **Sed aliquet**, neque at rutrum mollis, neque nisi tincidunt nibh, vitae faucibus lacus nunc at lacus. Nunc scelerisque, quam id cursus sodales, lorem libero fermentum urna, ut efficitur elit ligula et nunc.
>
> > Mauris dictum mi lacus, sit amet pellentesque urna vehicula fringilla. Ut sit amet placerat ante. Proin sed elementum nulla. Nunc vitae sem odio. Suspendisse ac eros arcu. Vivamus orci erat, volutpat a tempor et, rutrum. eu odio.
> >
> > > `Suspendisse rutrum facilisis risus`, eu posuere neque commodo a. Interdum et malesuada fames ac ante ipsum primis in faucibus. Sed nec leo bibendum, sodales mauris ut, tincidunt massa.

7.4.2 Other content blocks

> Vestibulum vitae orci quis ante viverra ultricies ut eget turpis. Sed eu lectus dapibus, eleifend nulla varius, lobortis turpis. In ac hendrerit nisl, sit amet laoreet nibh.

```
var _extends = function(target) {
  for (var i = 1; i < arguments.length; i++) {
    var source = arguments[i];
    for (var key in source) {
      target[key] = source[key];
    }
  }
  return target;
};
```

> > > Praesent at `:::js return target`, sodales nibh vel, tempor felis. Fusce vel lacinia lacus. Suspendisse rhoncus nunc non nisi iaculis ultrices. Donec consectetur mauris non neque imperdiet, eget volutpat libero.

7.5 Lists

7.5.1 Unordered lists

- Sed sagittis eleifend rutrum. Donec vitae suscipit est. Nullam tempus tellus non sem sollicitudin, quis rutrum leo facilisis. Nulla tempor lobortis orci, at elementum urna sodales vitae. In in vehicula nulla, quis ornare libero.
  - Duis mollis est eget nibh volutpat, fermentum aliquet dui mollis.
  - Nam vulputate tincidunt fringilla.
  - Nullam dignissim ultrices urna non auctor.
- Aliquam metus eros, pretium sed nulla venenatis, faucibus auctor ex. Proin ut eros sed sapien ullamcorper consequat. Nunc ligula ante, fringilla at aliquam ac, aliquet sed mauris.
- Nulla et rhoncus turpis. Mauris ultricies elementum leo. Duis efficitur accumsan nibh eu mattis. Vivamus tempus velit eros, porttitor placerat nibh lacinia sed. Aenean in finibus diam.

7.5.2 Ordered lists

1. Integer vehicula feugiat magna, a mollis tellus. Nam mollis ex ante, quis elementum eros tempor rutrum. Aenean efficitur lobortis lacinia. Nulla consectetur feugiat sodales.
2. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam ornare feugiat quam et egestas. Nunc id erat et quam pellentesque lacinia eu vel odio.
   1. Vivamus venenatis porttitor tortor sit amet rutrum. Pellentesque aliquet quam enim, eu volutpat urna rutrum a. Nam vehicula nunc mauris, a ultricies libero efficitur sed.
      1. Mauris dictum mi lacus
      2. Ut sit amet placerat ante
      3. Suspendisse ac eros arcu
   2. Morbi eget dapibus felis. Vivamus venenatis porttitor tortor sit amet rutrum. Pellentesque aliquet quam enim, eu volutpat urna rutrum a. Sed aliquet, neque at rutrum mollis, neque nisi tincidunt nibh.
   3. Pellentesque eget `:::js var _extends` ornare tellus, ut gravida mi.

```
var _extends = function(target) {
  for (var i = 1; i < arguments.length; i++) {
    var source = arguments[i];
    for (var key in source) {
      target[key] = source[key];
    }
  }
  return target;
};
```

3. Vivamus id mi enim. Integer id turpis sapien. Ut condimentum lobortis sagittis. Aliquam purus tellus, faucibus eget urna at, iaculis venenatis nulla. Vivamus a pharetra leo.

7.5.3 Definition lists

Not supported in commonmark, but you can use a rst definition list inside a fenced eval_rst block.

Lorem ipsum dolor sit amet Sed sagittis eleifend rutrum. Donec vitae suscipit est. Nullam tempus tellus non sem sollicitudin, quis rutrum leo facilisis. Nulla tempor lobortis orci, at elementum urna sodales vitae. In in vehicula nulla.

Duis mollis est eget nibh volutpat, fermentum aliquet dui mollis. Nam vulputate tincidunt fringilla. Nullam dignissim ultrices urna non auctor.

Cras arcu libero Aliquam metus eros, pretium sed nulla venenatis, faucibus auctor ex. Proin ut eros sed sapien ullamcorper consequat. Nunc ligula ante, fringilla at aliquam ac, aliquet sed mauris.

7.6 Code blocks

7.6.1 Inline

Morbi eget `dapibus felis`. Vivamus `venenatis porttitor` tortor sit amet rutrum. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. `Pellentesque aliquet quam enim`, eu volutpat urna rutrum a.

Nam vehicula nunc `:::js return target` mauris, a ultricies libero efficitur sed. Sed molestie imperdiet consectetur. Vivamus a pharetra leo. Pellentesque eget ornare tellus, ut gravida mi. Fusce vel lacinia lacus.

7.6.2 Listing

```
var _extends = function(target) {
  for (var i = 1; i < arguments.length; i++) {
    var source = arguments[i];
    for (var key in source) {
      target[key] = source[key];
    }
  }
  return target;
};
```

7.7 Horizontal rules

Aenean in finibus diam. Duis mollis est eget nibh volutpat, fermentum aliquet dui mollis. Nam vulputate tincidunt fringilla. Nullam dignissim ultrices urna non auctor.

Integer vehicula feugiat magna, a mollis tellus. Nam mollis ex ante, quis elementum eros tempor rutrum. Aenean efficitur lobortis lacinia. Nulla consectetur feugiat sodales.

7.8 Data tables

Note: Markdown table syntax requires `sphinx_markdown_tables`

Sed sagittis eleifend rutrum. Donec vitae suscipit est. Nullam tempus tellus non sem sollicitudin, quis rutrum leo facilisis. Nulla tempor lobortis orci, at elementum urna sodales vitae. In in vehicula nulla, quis ornare libero.

Vestibulum vitae orci quis ante viverra ultricies ut eget turpis. Sed eu lectus dapibus, eleifend nulla varius, lobortis turpis. In ac hendrerit nisl, sit amet laoreet nibh.

8 rst Cheatsheet

The rst Cheatsheet[12] covers a wide range of rst markup. It and its contents are CC licensed[13].

8.1 Inline Markup

Inline markup allows words and phrases within text to have character styles (like italics and boldface) and functionality (like hyperlinks).

[12] https://github.com/ralsina/rst-cheatsheet
[13] http://creativecommons.org/licenses/by/3.0/de/deed.en_GB

`*emphasis*`	emphasis
`**strong emphasis**`	**strong emphasis**
`` `interpreted text` ``	The rendering and meaning of interpreted text is domain- or application-dependent.
`` ``inline literal`` ```	`inline literal`
`reference_`	reference[14]
`` `phrase reference`_ ``	phrase reference[15]
`anonymous__`	anonymous[16]
`_`inline internal target``	inline internal target
`|substitution reference|`	The result is substituted in from the substitution definition.
`footnote reference [1]_`	footnote reference[17]
`citation reference [CIT2002]_`	citation reference [CIT2002]
`http://docutils.sf.net/`	http://docutils.sf.net/

[14] http://docutils.sourceforge.net/docs/user/rst/quickref.html#hyperlink-targets
[15] http://docutils.sourceforge.net/docs/user/rst/quickref.html#hyperlink-targets
[16] http://docutils.sourceforge.net/docs/user/rst/quickref.html#hyperlink-targets
[17] This is the first one.

8.2 Escaping with Backslashes

reStructuredText uses backslashes (“”) to override the special meaning given to markup characters and get the literal characters themselves. To get a literal backslash, use an escaped backslash (“₩”). For example:

escape ``with`` "\"	escape with “”
escape \``with`` "\\"	*escape* “with“ “”

8.3 Lists

```
- This is item 1. A blank line before
↪the first
  and last items is required.
- This is item 2

- Item 3: blank lines between items are
↪optional.
- Item 4: Bullets are "-", "*" or "+".
  Continuing text must be aligned after
↪the bullet
  and whitespace.
- This list item contains nested items

  - Nested items must be indented to
↪the same
    level
```

- This is item 1. A blank line before the first and last items is required.
- This is item 2
- Item 3: blank lines between items are optional.
- Item 4: Bullets are “-“, “*” or “+”. Continuing text must be aligned after the bullet and whitespace.
- This list item contains nested items
  - Nested items must be indented to the same level

```
3. This is the first item
4. This is the second item
5. Enumerators are arabic numbers,
   single letters, or roman numerals
6. List items should be sequentially
   numbered, but need not start at 1
   (although not all formatters will
   honour the first index).
#. This item is auto-enumerated
```

3. This is the first item
4. This is the second item
5. Enumerators are arabic numbers, single letters, or roman numerals
6. List items should be sequentially numbered, but need not start at 1 (although not all formatters will honour the first index).
7. This item is auto-enumerated

```
what
  Definition lists associate a term with
  a definition.

how
  The term is a one-line phrase, and the
  definition is one or more paragraphs or
  body elements, indented relative to the
  term. Blank lines are not allowed
  between term and definition.
```

what Definition lists associate a term with a definition.

how The term is a one-line phrase, and the definition is one or more paragraphs or body elements, indented relative to the term. Blank lines are not allowed between term and definition.

```
:Authors:
    Tony J. (Tibs) Ibbs,
    David Goodger

:Version: 1.0 of 2001/08/08
:Dedication: To my father.
```

Authors Tony J. (Tibs) Ibbs, David Goodger
Version 1.0 of 2001/08/08
Dedication To my father.

```
-a            command-line option "a"
-b file       options can have arguments
              and long descriptions
--long        options can be long also
--input=file  long options can also have
              arguments
/V            DOS/VMS-style options too
```

-a	command-line option “a”
-b file	options can have arguments and long descriptions
--long	options can be long also
--input=file	long options can also have arguments

8.4 Section Structure

```
Title
=====

Titles are underlined (or over- and
↪underlined) with
a nonalphanumeric character at least as
↪long as the
text.

A lone top-level section is lifted up
↪to be the
document's title.

Any non-alphanumeric character can be
↪used, but
Python convention is:

* ``#`` with overline, for parts
* ``*`` with overline, for chapters
* ``=``, for sections
* ``-``, for subsections
* ``^``, for subsubsections
* ``"``, for paragraphs
```

Title

Titles are underlined (or over- and underlined) with a nonalphanumeric character at least as long as the text.

A lone top-level section is lifted up to be the document's title.

Any non-alphanumeric character can be used, but Python convention is:

- # with overline, for parts
- * with overline, for chapters
- =, for sections
- -, for subsections
- ^, for subsubsections
- ", for paragraphs

8.5 Blocks

```
This is a paragraph.

Paragraphs line up at their left edges,
↪and are
normally separated by blank lines.
```

This is a paragraph.
Paragraphs line up at their left edges, and are normally separated by blank lines.

```
A paragraph containing only two colons
↪indicates
the following indented or quoted text
↪is a literal
block or quoted text is a literal block.

::

  Whitespace, newlines, blank lines,
↪and  all kinds of
  markup (like *this* or \this) is
↪preserved here.

You can also tack the ``::`` at the end
↪of a
paragraph::

   It's very convenient to use this form.

Per-line quoting can also be used for
↪unindented
blocks::

> Useful for quotes from email and
> for Haskell literate programming.
```

A paragraph containing only two colons indicates that the following indented or quoted text is a literal block.

```
Whitespace, newlines, blank lines, and
all kinds of markup (like *this* or
\this) is preserved by literal blocks.
```

You can also tack the : : at the end of a paragraph:

```
It's very convenient to use this form.
```

Per-line quoting can also be used for unindented blocks:

```
> Useful for quotes from email and
> for Haskell literate programming.
```

```
| Line blocks are useful for addresses,
| verse, and adornment-free lists.
|
| Each new line begins with a
| vertical bar ("|").
|     Line breaks and initial indents
|     are preserved.
| Continuation lines are wrapped
  portions of long lines; they begin
  with spaces in place of vertical bars.
```

Line blocks are useful for addresses,
verse, and adornment-free lists.

Each new line begins with a
vertical bar (“|”).
Line breaks and initial indents
are preserved.
Continuation lines are wrapped portions of long lines; they begin with spaces in place of vertical bars.

```
Block quotes are just:

    Indented paragraphs,

        and they may nest.
```

Block quotes are just:

Indented paragraphs,

and they may nest.

```
Doctest blocks are interactive
Python sessions. They begin with
"``>>>``" and end with a blank line.

>>> print "This is a doctest block."
This is a doctest block.
```

Doctest blocks are interactive Python sessions. They begin with “>>>” and end with a blank line.

```
>>> print "This is a doctest block."
This is a doctest block.
```

8.6 Tables

There are two syntaxes for tables in reStructuredText. Grid tables are complete but cumbersome to create. Simple tables are easy to create but limited (no row spans, etc.).

```
+------------+------------+-----------+
| Header 1   | Header 2   | Header 3  |
+============+============+===========+
| body row 1 | column 2   | column 3  |
+------------+------------+-----------+
| body row 2 | Cells may span columns.|
+------------+------------+-----------+
| body row 3 | Cells may  | - Cells   |
+------------+ span rows. | - contain |
| body row 4 |            | - blocks. |
+------------+------------+-----------+
```

Header 1	Header 2	Header 3
body row 1	column 2	column 3
body row 2	Cells may span columns.	
body row 3	Cells may span rows.	• Cells • contain • blocks.
body row 4		

```
=====  =====  ======
   Inputs     Output
------------  ------
  A      B    A or B
=====  =====  ======
False  False  False
True   False  True
False  True   True
True   True   True
=====  =====  ======
```

Inputs		Output
A	B	A or B
False	False	False
True	False	True
False	True	True
True	True	True

8.7 Explicit Markup

Explicit markup blocks are used for constructs which float (footnotes), have no direct paper-document representation (hyperlink targets, comments), or require specialized processing (directives). They all begin with two periods and whitespace, the “explicit markup start”.

```
Footnote references, like [5]_.
Note that footnotes may get
rearranged, e.g., to the bottom of
the "page".

.. [5] A numerical footnote. Note
   there's no colon after the ``]``.
```

Footnote references, like[5]. Note that footnotes may get rearranged, e.g., to the bottom of the “page”.

```
Autonumbered footnotes are
possible, like using [#]_ and [#]_.

.. [#] This is the first one.
.. [#] This is the second one.

They may be assigned 'autonumber
labels' - for instance,
[#fourth]_ and [#third]_.

.. [#third] a.k.a. third_

.. [#fourth] a.k.a. fourth_
```

Autonumbered footnotes are possible, like using[17] and[18].
They may be assigned ‘autonumber labels’ - for instance,[20] and[19].

```
Auto-symbol footnotes are also
possible, like this: [*]_ and [*]_.

.. [*] This is the first one.
.. [*] This is the second one.
```

Auto-symbol footnotes are also possible, like this:[21] and[22].

```
Citation references, like [CIT2002]_.
Note that citations may get
rearranged, e.g., to the bottom of
the "page".

.. [CIT2002] A citation
   (as often used in journals).

Citation labels contain alphanumerics,
underlines, hyphens and fullstops.
Case is not significant.

Given a citation like [this]_, one
can also refer to it like this_.

.. [this] here.
```

Citation references, like [CIT2002]. Note that citations may get rearranged, e.g., to the bottom of the “page”.
Citation labels contain alphanumerics, underlines, hyphens and fullstops. Case is not significant.
Given a citation like [this], one can also refer to it like this.

```
External hyperlinks, like Python_.

.. _Python: http://www.python.org/
```

External hyperlinks, like Python[23].

```
External hyperlinks, like `Python
<http://www.python.org/>`_.
```

External hyperlinks, like Python[24].

```
Internal crossreferences, like example_.

.. _example:

This is an example crossreference target.
```

Internal crossreferences, like example.
This is an example crossreference target.

8.8 Credits

CP Font from LiquiType:	http://www.liquitype.com/workshop/type_design/cp-mono
Magnetic Balls V2 image by fdecomite:	http://www.flickr.com/photos/fdecomite/2926556794/
Sponsored by Net Managers	http://www.netmanagers.com.ar
Typeset using rst2pdf	http://rst2pdf.googlecode.com

© 2009 Roberto Alsina ⟨ralsina@netmanagers.com.ar⟩ / Creative Commons Attribution-Noncommercial-Share Alike 2.5 Argentina License	Based on quickref.txt from docutils	Non-Commercial	Share Alike

[5] A numerical footnote. Note there's no colon after the].
[18] This is the second one.
[20] a.k.a. fourth
[19] a.k.a. third
[21] This is the first one.
[22] This is the second one.
[23] http://www.python.org/
[24] http://www.python.org/
[25] http://www.python.org/
[26] http://www.python.org/

9 Change Log

Warning: Changes are not being tracked until a beta-quality release is made.

The change log will appear here.

10

License

Copyright (c) 2019 Kevin Sheppard ⟨kevin.k.sheppard@gmail.com⟩

Derived from:

- Material for Mkdocs: Copyright (c) 2016-2019 Martin Donath ⟨martin.donath@squidfunk.com⟩
- Guzzle Sphinx Theme: Copyright (c) 2013 Michael Dowling ⟨mtdowling@gmail.com⟩

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

11

Index

genindex

Bibliography

[CIT2002] A citation (as often used in journals).
[this] here.

Index