# Writing and running efficient Perl on Genepool

## version control, analysis, packaging, and environment modules

GENE DESIGN

BioStudio

CloneQC

bitbucket.org/notadoctor

search.cpan.org/~notdoctor/

# Excellent Practices

Version Control

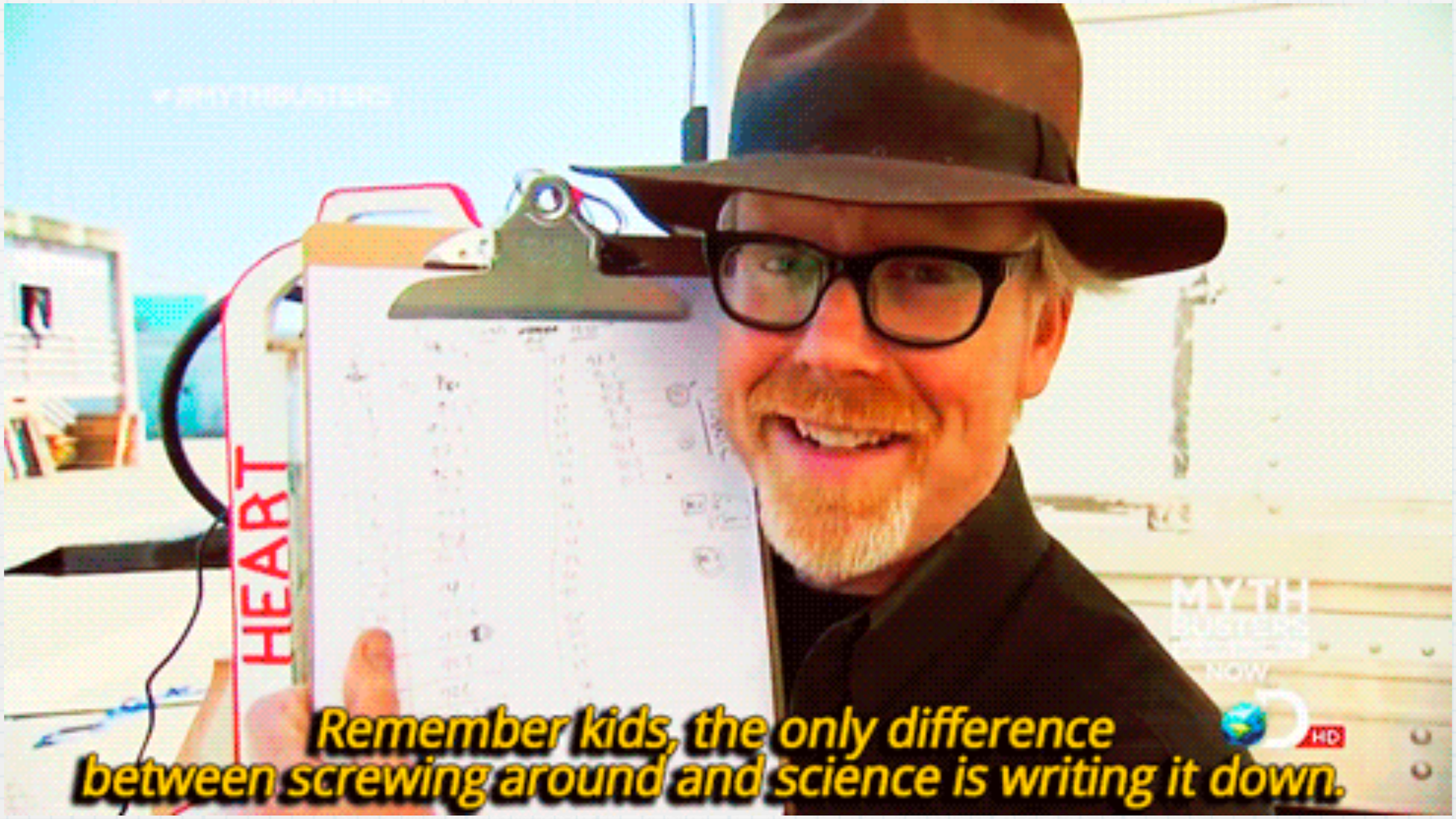
Dynamic Analysis - Profiling

Static Analysis - Criticism

Packaging and Testing

Environment Modules

# Version Control




# the coder's laboratory notebook

notadoctor / GeneDesign-4

Atlassian, Inc. [US] https://bitbucket.org/notadoctor/genedesign-4/commits/all/069029baed36

| | | | |
|---|---|---|---|
| Sarah Richard... | fc13eca | fixed algorithm bugs | 2009-12-14 |
| Sarah Richard... | 06d5d00 | Pruning SC2.0 specific code | 2009-12-14 |
| Sarah Richard... | 7b9215d | tweaks | 2009-12-14 |
| Sarah Richard... | d07113f | Added use warnings; new enzyme list; FASTA friendliness; renamed libraries | 2009-12-14 |
| Sarah Richard... | 8816aae | more cpan dependencies | 2009-10-19 |
| Sarah Richard... | 075add0 | I have no idea why User wasn't checked more carefully; should be generating aligned oligo | 2009-10-18 |
| Sarah Richard... | 393ef24 | Updated PML.pm | 2009-10-18 |
| Sarah Richard... | 41c9f40 | Added info on cpan prereqs | 2009-10-09 |
| Sarah Richard... | bd647cd | fixed oligo code in GeneDesign.pm causing hash errors for gdUserDes | 2009-10-09 |
| Sarah Richard... | 3eeff67 | Fixed regular expression to allow chunks to have double digit names | 2009-10-09 |
| Sarah Richard... | 618eb1a | Ready for installation I think | 2009-08-18 |
| Sarah Richard... | 6188020 | updating the html and css files | 2009-08-18 |
| Sarah Richard... | 021766d | oops | 2009-08-18 |
| Sarah Richard... | 243d877 | bringing it up to date | 2009-08-18 |
| Sarah Richard... | c4c5204 | More code brought up to date | 2009-08-13 |
| Sarah Richard... | dbacf64 | Bringing it up to date | 2009-08-11 |
| Sarah Richard... | 503ce5a | Bringing it up to date | 2009-08-11 |
| Sarah Richard... | 36bde10 | checking branch | 2009-08-06 |
| Sarah Richard... | 6576ec9 | These files were deployed on macbeth.clark.jhu.edu as genedesign.org; This is the reference | 2009-08-06 |
| Sarah Richard... | 15324cd | published version; GeneDesign 2.0 | 2009-08-06 |

JGISynBio — JGISB Build — less — 158×43

GD Build | GD Run | CQC Build bash... | CQC Run bash... | JGISB Build

```
richards:JGISynBio richards$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   bin/JGISB_Order_Stitch.pl
#       modified:   bin/JGISB_SubmitOligoJob.pl
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       JGISB_submit.sh
no changes added to commit (use "git add" and/or "git commit -a")
richards:JGISynBio richards$
richards:JGISynBio richards$
richards:JGISynBio richards$ git diff bin/JGISB_Order_Stitch.pl
diff --git a/bin/JGISB_Order_Stitch.pl b/bin/JGISB_Order_Stitch.pl
index f39065a..55f8e8a 100755
--- a/bin/JGISB_Order_Stitch.pl
+++ b/bin/JGISB_Order_Stitch.pl
@@ -1,10 +1,7 @@
 #!/usr/bin/env perl

 use Bio::GeneDesign;
-use Spreadsheet::ParseExcel;
 use Spreadsheet::WriteExcel;
-use File::Basename;
-use Config::Auto;
 use Getopt::Long;
 use Pod::Usage;

@@ -22,11 +19,9 @@ $| = 1;
```

genedesign-dev – Commits (42 of 42 Commits)

Repositories | Fetch | Pull | Push | Commit | Merge | Rebase | Save Stash | Apply Stash | Refresh | Settings | Search History

BRANCHES
master...
TAGS
REMOTES
origin
ma...
STASHES
SUBMODULES

Status (2) | Commits | Browse

Filters | Load More | Recent | List

| Date | Subject | Author | Hash |
|---|---|---|---|
| February 9 | add the pattern aligner to the engine :/ | Sarah Richardson | ce37b3826021d7c5b0d9fa4a264e33cbb866c4d0 |
| February 9 | Merge branch 'master' of https://bitbucket.org/notadoctor/genedesign-dev | Sarah Richardson | 2e6bc9c36dc1bec9bb31d50cf2c01dd0c00b9eca |
| February 9 | adjust suffix tree so that in peptide implementations, the originating nucle... | Sarah Richardson | eec4f9022bb0f6d11e8649f787e2b6b246081cf4 |
| February 7 | when Bio::ConfigData isn't installed, this line causes an uninitialized warning | notadoctor | 49d525929225336892f77e76d98abe992b0a88d8 |
| February 7 | if running in simple mode, will NOT have enz/zne tags to be removed. | Sarah Richardson | 23353c236067cb3976ce830c7a4dfe7fa21dce5f |
| February 7 | make sure sequences are uppercase | Sarah Richardson | ac6eb758502e57166e0ac4096cae30d17cf7da98 |
| February 7 | oops - pass the clonvec object, not a reference to the object | Sarah Richardson | faefcf6860c468bbad41cbd8dfca05c4dd3709c9 |
| February 5 | add pFIL to vector list | Sarah Richardson | ed287a2531caa836fd635ba19400a10b2e264dbe |
| February 5 | fix non ASCII character | Sarah Richardson | f6af8065ff224f077719795999217ac2854a16b3 |
| February 5 | fix pod test non-ascii character error | Sarah Richardson | a513355b92a0d705f6775eea40a7f61ec9e50b27 |
| February 2 | Add POD coverage for _checkref | Sarah Richardson | 0b7756237e7de0d73d9e9d0a4773112f1952ada7 |
| February 1 | Add a stripdown function | Sarah Richardson | 59710aab80b657da9b6b2c0a8b17fac7607bc005 |

SHA ac6eb758502e57166e0ac4096cae30d17cf7da98
AUTHOR Sarah Richardson
DATE February 7, 2013 4:08 PM
SUBJECT **make sure sequences are uppercase**
PARENT faefcf6860c468bbad41cbd8dfca05c4dd3709c9
TREE 055a43c2c93bcadee3c513f4cc97bda49f41cb74

**make sure sequences are uppercase**

modified M lib/Bio/GeneDesign/Vector.pm

**lib/Bio/GeneDesign/Vector.pm**

2 chunks, 2 insertions, 2 deletions

```
@@ -66,7 +66,7 @@ sub new
66 66   my %dsubs = map {join(q{}, $_->get_tag_values("label")) => $_} @ps;
67 67   if (exists $dsubs{"CLH5"})
68 68   {
69    -   $self->{chew5} = $dsubs{"CLH5"}->seq->seq;
   69 +   $self->{chew5} = uc $dsubs{"CLH5"}->seq->seq;
70 70   }
71 71   else
72 72   {
@@ -75,7 +75,7 @@ sub new
75 75   }
76 76   if (exists $dsubs{"CLH3"})
77 77   {
78    -   $self->{chew3} = $dsubs{"CLH3"}->seq->seq;
   78 +   $self->{chew3} = uc $dsubs{"CLH3"}->seq->seq;
```

# Version Control Carrots

**branching:** risk free development while your production code ticks onwards unmolested. Take the fear out of refactoring!

**backups:** a git server and a healthy commit habit let you work from anywhere - and can prevent you from losing work with an unconsidered overwrite.

**blamestorming:** in groups (n >= 1) it is easy to find out who is contributing (or breaking) what, or to organize an issue system to assign tasks or track bugs tied to commits.

**ninja deploys:** cron jobs can automatically pull the latest version of your code from the repository and deploy it on genepool while you eat lunch or nap!

**...and so many more**

# Version Control Sticks


If you can think of a single situation in which not using version control is an entirely logical and defensible position email it to me at SMRichardson

I'm dying to hear it

# git started now!

LBNL on bitbucket
https://commons.lbl.gov/display/itfaq/Version+Control+Systems

NERSC on git
http://www.nersc.gov/users/software/development-tools/git/

a free, easy to read, and all around wonderful
introduction to source control and git reference
http://git-scm.com/book

do right by your code.

**git init my_code_dir/**

# Dynamic Analysis

```
#
# This is O(scary), but seems quick enough in practice.
#
foreach my $a (@bigarray)
{
  foreach my $b (@biggerarray)
  {
    foreach my $c (@youdontwannaknow)
    {
      foreach my $d (@justgiveup)
      {
        push @newarray, function($a, $b, $c, $d);
      }
    }
  }
}
```

**check yourself before you wreck yourself**

makingoligos — GD Run — bash — 110×24

GD Build | GD Run | CQC Build ... | CQC Run ... | JGISB Build | JGISB Run ... | BS Build ... | BS Run b...

```
richards:makingoligos richards$ time -p GD_Design_Building_Blocks.pl -i Test_one_big.FASTA -o . -f genbank
Working on Monocot_A1 (6125 bp)...
Warning: integer value out of range 20 more than (reset to) 14


Made building blocks of Monocot_A1

Wrote ./Test_one_big_BB.genbank

Awkwardly brought to you by GD_Design_Building_Blocks_5.00

real 23.87
user 12.20
sys 8.79
richards:makingoligos richards$
```

# OPEN

**All the Code That's Fit to `printf()`**

March 5, 2008, 1:43 pm | 13 Comments

## The New York Times Perl Profiler

By ADAM KAPLAN

I work in the NYTimes.com feeds team. We handle retrieving, parsing and transforming incoming feeds from whatever strange proprietary format our partners choose to give us into something that our CMS can digest. As you can imagine, we deal with a huge amount of text processing. To handle all of these transformations as efficiently as possible we rely heavily on the magic of Perl. Recently, as feeds become more and more important, we have begun to feel pains caused by past impromptu segments of inefficient code written to meet quick, episodic deadlines. A situation that we are especially prone to as a fast moving news organization.

I am a relatively new employee here at NYTimes.com and one of my responsibilities is to create tools to help ensure the integrity and scalability of our code. To this end, I would like to introduce you to The New York Times Perl Profiler, or Devel::NYTProf. The purpose of this tool is to allow developers to easily profile Perl code line-by-line with minimal computational overhead and highly visual output. With only one additional command, developers can generate robust color-coded HTML reports that include some useful statistics about their Perl program. Here is the typical usage:

```
perl -d:NYTProf myslowcode.pl
nytprofhtml
```

See? It's easy! nytprofhtml is an implementation of the included reporting

file://localhost/Users/richards/test/nytprof/index.html

# Performance Profile Index

For /usr/local/bin/BS_ChromosomeAnalyzer.pl

Run on Tue Feb 12 00:09:53 2013
Reported on Tue Feb 12 00:11:35 2013

Profile of /usr/local/bin/BS_ChromosomeAnalyzer.pl for 33.4s (of 51.1s), executing 37709980 statements and 2228909 subroutine calls in 230 source files and 76 string evals.

/Library/Perl/5.12/Bio/AnalysisParserI.pm

## Top 15 Subroutines

| Calls | P | F | Exclusive Time | Inclusive Time | Subroutine |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 7.41s | 7.76s | Bio::BioStudio::Mask::find_overlaps |
| 1 | 1 | 1 | 4.46s | 4.81s | Bio::BioStudio::Mask::find_deserts |
| 27282 | 9 | 2 | 4.22s | 4.22s | DBI::st::execute (xsub) |
| 1 | 1 | 1 | 3.42s | 3.45s | Bio::BioStudio::Mask::add_to_mask |
| 1 | 1 | 1 | 1.31s | 1.31s | Bio::BioStudio::Mask::new |
| 1 | 1 | 1 | 984ms | 32.1s | Bio::BioStudio::Analyze::ProteinCodingGenes::_analyze |
| 1 | 1 | 1 | 917ms | 1.27s | Bio::GeneDesign::Codons::_codon_count |
| 18240 | 1 | 1 | 445ms | 5.32s | Bio::DB::SeqFeature::Store::DBI::mysql::_fetch_SeqFeatures |
| 3266 | 3 | 1 | 353ms | 353ms | Bio::GeneDesign::Codons::CORE:match (opcode) |
| 82201 | 7 | 1 | 321ms | 448ms | Bio::DB::SeqFeature::Store::DBI::mysql::_qualify |
| 69945 | 5 | 2 | 305ms | 917ms | Bio::SeqFeature::Generic::start |
| 69945 | 5 | 2 | 294ms | 831ms | Bio::SeqFeature::Generic::end |
| 69945 | 1 | 1 | 291ms | 439ms | Bio::Location::Simple::end |
| 27281 | 1 | 1 | 283ms | 398ms | DBD::_::db::prepare_cached |
| 73827 | 2 | 2 | 276ms | 403ms | Bio::Location::Simple::start |

See all 4273 subroutines

You can view a treemap of subroutine exclusive time, grouped by package.
NYTProf also generates call-graph files in Graphviz format: inter-package calls, all inter-subroutine calls (probably too complex to render easily).

You can hover over some table cells and headings to view extra information.
Some table column headings can be clicked on to sort the table by that column.

file://localhost/Users/richards/test/nytprof/Bio-BioStudio-Mask-pm-193-line.html#142

| | | | | | |
|---|---|---|---|---|---|
| 140 | | | | | |
| 141 | | | | | sub find_overlaps |
| 142 | | | | | # spent 7.76s (7.41+358ms) within Bio::BioStudio::Mask::find_overlaps which was called:<br># once (7.41s+358ms) by Bio::BioStudio::Analyze::ProteinCodingGenes::_analyze at line 206 of Bio/B<br>{ |
| 143 | 1 | 700ns | | | my ($self) = @_; |
| 144 | 1 | 100ns | | | my @ranges; |
| 145 | | | | | |
| 146 | 1 | 900ns | | | my $len = $self->{length}; |
| 147 | 1 | 2µs | | | my $init  = $self->{mask}->[0]->[0]; |
| 148 | 1 | 700ns | | | my $start = $init > 1 ? $init : undef; |
| 149 | 1 | 600ns | | | my $last  = $init > 1 ? $init : 1; |
| 150 | 1 | 300ns | | | my $flag  = $init > 1 ? $init : 0; |
| 151 | | | | | |
| 152 | 1 | 800ns | | | my %list = (); |
| 153 | 1 | 743ms | | | for (my $x = 0; $x < $len; $x++) |
| 154 | | | | | { |
| 155 | 3282708 | 2.56s | | | my @here = @{$self->{mask}->[$x]}; |
| 156 | 3282708 | 348ms | | | my $stat = shift @here; |
| 157 | | | | | |
| 158 | | | | | #moving from non overlap to overlap |
| 159 | 3282708 | 668ms | | | if ($stat > 1 && $last <= 1) |
| 160 | | | | | { |
| 161 | 471 | 182µs | | | $start = $x+1; |
| 162 | 471 | 48µs | | | $flag = 1; |
| 163 | 942 | 2.27ms | | | $list{$_} = [$x+1, $x+1] foreach @here; |
| 164 | | | | | } |
| 165 | | | | | #moving from overlap to non overlap |
| 166 | | | | | elsif ($last > 1 && $stat <= 1) |
| 167 | | | | | { |
| 168 | 471 | 338µs | | | my $seqid = $self->{seqid}; |
| 169 | 471 | 2.53ms | 471 | 90.7ms | my $lapfeat = Bio::SeqFeature::Generic->new(<br># spent 90.7ms making 471 calls to Bio::SeqFeature::Generic::new, avg 193µs/call |
| 170 | | | | | -start => $start, |
| 171 | | | | | -end => $x, |
| 172 | | | | | -display_name => "$start..$x", |
| 173 | | | | | -primary_tag => "overlap", |
| 174 | | | | | -seq_id => $seqid |
| 175 | | | | | ); |
| 176 | 471 | 855µs | | | foreach my $featid (keys %list) |
| 177 | | | | | { |
| 178 | 942 | 539µs | | | my ($fstart, $fstop) = @{$list{$featid}}; |
| 179 | 942 | 4.59ms | 942 | 149ms | my $subfeat = Bio::SeqFeature::Generic->new(<br># spent 149ms making 942 calls to Bio::SeqFeature::Generic::new, avg 158µs/call |

reddit: the front page of the | Profile of Bio/BioStudio/Mas | Profile of Bio/BioStudio/Mas | Profile of Bio/BioStudio/Mas

file://localhost/Users/richards/test/nytprof/Bio-BioStudio-Mask-pm-193-line.html#144

| | | | | | |
|---|---|---|---|---|---|
| 143 | | | | | sub find_overlaps |
| 144 | | | | | # spent 4.51s (4.13+384ms) within Bio::BioStudio::Mask::find_overlaps which was called:<br># once (4.13s+384ms) by Bio::BioStudio::Analyze::ProteinCodingGenes::_analyze at line 206 of Bio/B<br>{ |
| 145 | 1 | 800ns | | | my ($self) = @_; |
| 146 | 1 | 100ns | | | my @ranges; |
| 147 | | | | | |
| 148 | 1 | 1µs | | | my $len = $self->{length}; |
| 149 | 1 | 900ns | | | my $mask = $self->{mask}; |
| 150 | 1 | 900ns | | | my $seqid = $self->{seqid}; |
| 151 | 1 | 2µs | | | my $init = $mask->[0]->[1]; |
| 152 | 1 | 600ns | | | my $start = $init > 1 ? $init : undef; |
| 153 | 1 | 500ns | | | my $last = $init > 1 ? $init : 1; |
| 154 | 1 | 200ns | | | my $flag = $init > 1 ? $init : 0; |
| 155 | | | | | |
| 156 | 1 | 600ns | | | my %list = (); |
| 157 | 1 | 741ms | | | for (my $x = 0; $x < $len; $x++) |
| 158 | | | | | { |
| 159 | 3282708 | 728ms | | | my $stat = $mask->[$x]->[1]; |
| 160 | | | | | |
| 161 | | | | | #moving from non overlap to overlap |
| 162 | 3282708 | 661ms | | | if ($stat > 1 && $last <= 1) |
| 163 | | | | | { |
| 164 | 471 | 265µs | | | $start = $x+1; |
| 165 | 471 | 67µs | | | $flag = 1; |
| 166 | 942 | 6.35ms | | | $list{$_} = [$x+1, $x+1] foreach keys %{$mask->[$x]->[0]}; |
| 167 | | | | | } |
| 168 | | | | | #moving from overlap to non overlap |
| 169 | | | | | elsif ($last > 1 && $stat <= 1) |
| 170 | | | | | { |
| 171 | 471 | 3.52ms | 471 | 102ms | my $lapfeat = Bio::SeqFeature::Generic->new(<br># spent 102ms making 471 calls to Bio::SeqFeature::Generic::new, avg 217µs/call |
| 172 | | | | | -start => $start, |
| 173 | | | | | -end => $x, |
| 174 | | | | | -display_name => "$start..$x", |
| 175 | | | | | -primary_tag => "overlap", |
| 176 | | | | | -seq_id => $seqid |
| 177 | | | | | ); |
| 178 | 471 | 855µs | | | foreach my $featid (keys %list) |
| 179 | | | | | { |
| 180 | 942 | 637µs | | | my ($fstart, $fstop) = @{$list{$featid}}; |
| 181 | 942 | 5.08ms | 942 | 157ms | my $subfeat = Bio::SeqFeature::Generic->new(<br># spent 157ms making 942 calls to Bio::SeqFeature::Generic::new, avg 166µs/call |

# Performance Profile Index

For /usr/local/bin/BS_ChromosomeAnalyzer.pl

Run on Tue Feb 12 01:26:21 2013
Reported on Tue Feb 12 01:28:08 2013

Profile of /usr/local/bin/BS_ChromosomeAnalyzer.pl for 31.1s (of 49.8s), executing 34426805 statements and 2228909 subroutine calls in 230 source files and 76 string evals.

/Library/Perl/5.12/Bio/AnalysisParserI.pm

## Top 15 Subroutines

| Calls | P | F | Exclusive Time | Inclusive Time | Subroutine |
|---|---|---|---|---|---|
| 27282 | 9 | 2 | 4.34s | 4.34s | DBI::st::execute (xsub) |
| 1 | 1 | 1 | 4.33s | 4.71s | Bio::BioStudio::Mask::find_overlaps |
| 1 | 1 | 1 | 4.20s | 4.58s | Bio::BioStudio::Mask::find_deserts |
| 1 | 1 | 1 | 2.85s | 2.89s | Bio::BioStudio::Mask::add_to_mask |
| 1 | 1 | 1 | 2.08s | 2.08s | Bio::BioStudio::Mask::new |
| 1 | 1 | 1 | 1.38s | 29.9s | Bio::BioStudio::Analyze::ProteinCodingGenes::_analyze |
| 1 | 1 | 1 | 935ms | 1.29s | Bio::GeneDesign::Codons::_codon_count |
| 18240 | 1 | 1 | 465ms | 5.50s | Bio::DB::SeqFeature::Store::DBI::mysql::_fetch_SeqFeatures |
| 3266 | 3 | 1 | 359ms | 359ms | Bio::GeneDesign::Codons::CORE:match (opcode) |
| 82201 | 7 | 1 | 327ms | 458ms | Bio::DB::SeqFeature::Store::DBI::mysql::_qualify |
| 69945 | 5 | 2 | 318ms | 944ms | Bio::SeqFeature::Generic::start |
| 69945 | 5 | 2 | 315ms | 864ms | Bio::SeqFeature::Generic::end |
| 69945 | 1 | 1 | 301ms | 452ms | Bio::Location::Simple::end |
| 27281 | 1 | 1 | 298ms | 415ms | DBD::_::db::prepare_cached |
| 73827 | 2 | 2 | 283ms | 414ms | Bio::Location::Simple::start |

See all 4273 subroutines

You can view a treemap of subroutine exclusive time, grouped by package.
NYTProf also generates call-graph files in Graphviz format: inter-package calls, all inter-subroutine calls (probably too complex to render easily).

You can hover over some table cells and headings to view extra information.
Some table column headings can be clicked on to sort the table by that column.

reddit: the fro | Profile of Bio/ | Profile of Bio/ | Profile of Bio/ | Profile of Bio/ | NYTProf BS_C | Profile of Bio/ | Subroutine Ex

file://localhost/Users/richards/test/nytprof/subs-treemap-excl.html

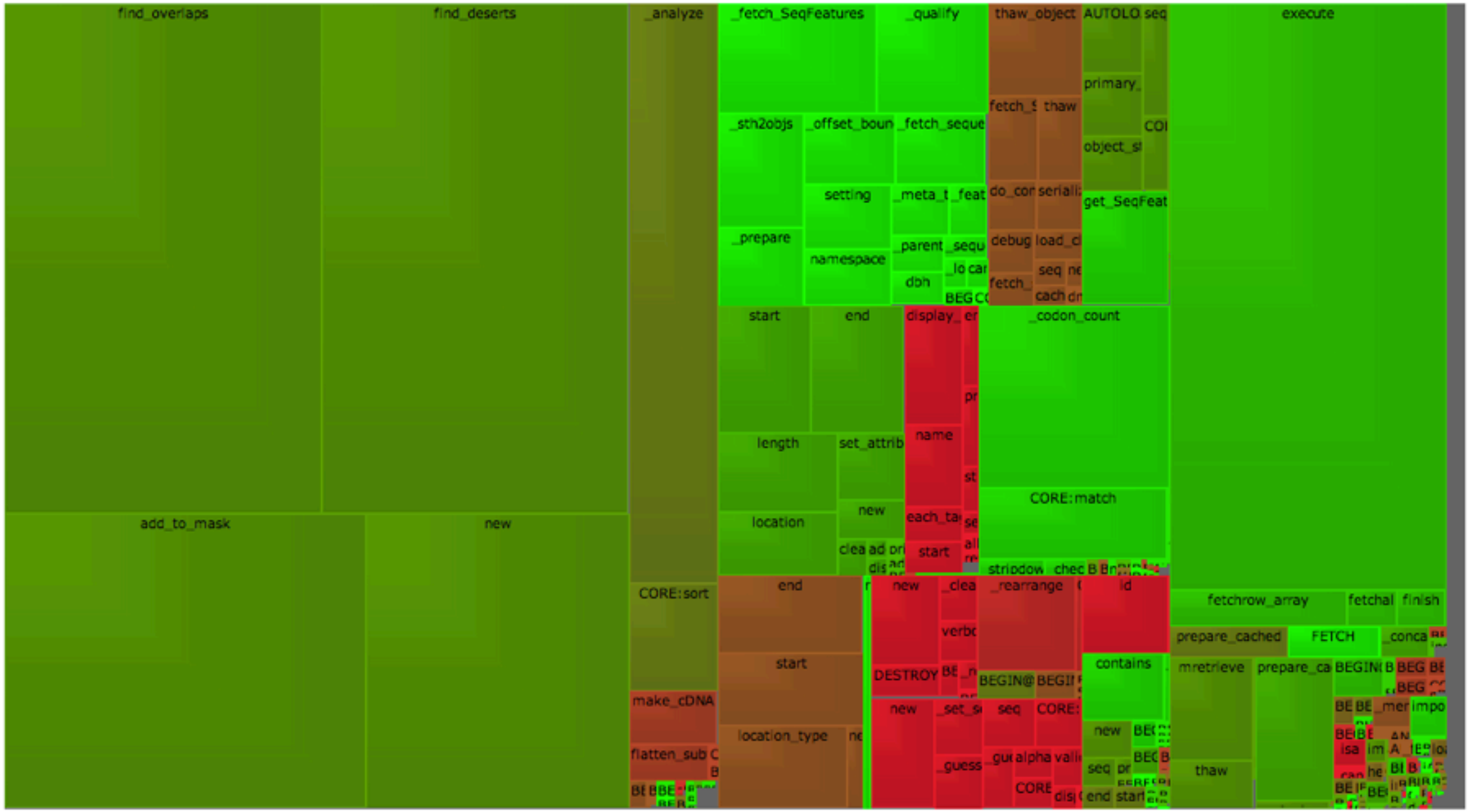
← Index

# Subroutine Exclusive Time Treemap

For /usr/local/bin/BS_ChromosomeAnalyzer.pl

Run on Tue Feb 12 01:26:21 2013
Reported on Tue Feb 12 01:28:08 2013

*Boxes represent time spent in a subroutine. Coloring represents packages. Click to drill-down into package hierarchy, reload page to reset.*




Report produced by the NYTProf 4.25 Perl profiler, developed by Tim Bunce and Adam Kaplan.

# profile away

NYTProfiler on CPAN - all the configuration options you could want
http://search.cpan.org/~timb/Devel-NYTProf-4.25/

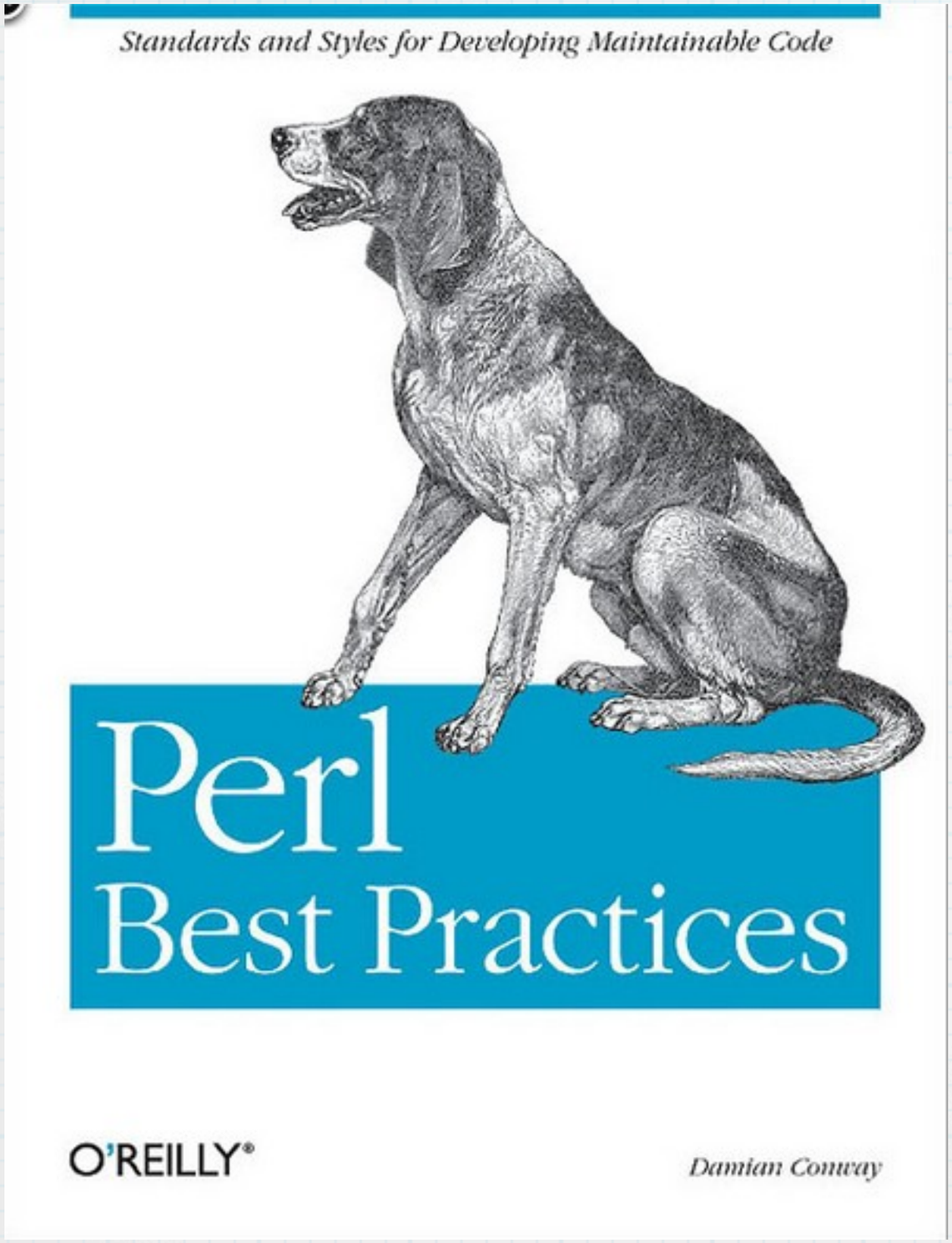
an excellent lecture by the maintainer of the NYTProfiler
http://blip.tv/timbunce/nytprof-v3-ipw-2009-2860609

are you O(scary) or O(awesome)?

```
perl -d:NYTProf your_code.pl
nytprofhtml
```

# Static Analysis


everyone's a critic




## 1. Best Practices

### Three Goals

- Robustness
- Efficiency
- Maintainability

biostudio-dev — BS Build — bash —

GD Build | GD Run | CQC Build | CQC Run | JGISB Build

```
richards:biostudio-dev richards$ perlcritic lib/Bio/BioStudio/Mask.pm
lib/Bio/BioStudio/Mask.pm source OK
richards:biostudio-dev richards$
```

GD Build | GD Run | CQC Build | CQ

```
richards:biostudio-dev richards$ perlcritic --harsh lib/Bio/BioStudio/Mask.pm
Ambiguously named variable "last" at line 113, column 3.  See page 48 of PBP.  (Severity: 3)
Ambiguously named variable "last" at line 153, column 3.  See page 48 of PBP.  (Severity: 3)
Subroutine name is a homonym for builtin function length at line 336, column 1.  See page 177 o
richards:biostudio-dev richards$ perlcritic --cruel lib/Bio/BioStudio/Mask.pm
Missing "SYNOPSIS" section in POD at line 5, column 1.  See pages 133,138 of PBP.  (Severity: 2
Missing "SUBROUTINES/METHODS" section in POD at line 5, column 1.  See pages 133,138 of PBP.  (
Missing "DIAGNOSTICS" section in POD at line 5, column 1.  See pages 133,138 of PBP.  (Severity
Missing "CONFIGURATION AND ENVIRONMENT" section in POD at line 5, column 1.  See pages 133,138 
Missing "DEPENDENCIES" section in POD at line 5, column 1.  See pages 133,138 of PBP.  (Severit
Missing "INCOMPATIBILITIES" section in POD at line 5, column 1.  See pages 133,138 of PBP.  (Se
Missing "BUGS AND LIMITATIONS" section in POD at line 5, column 1.  See pages 133,138 of PBP.  
Missing "LICENSE AND COPYRIGHT" section in POD at line 5, column 1.  See pages 133,138 of PBP.
Postfix control "unless" used at line 54, column 5.  See pages 96,97 of PBP.  (Severity: 2)
Postfix control "unless" used at line 57, column 4.  See pages 96,97 of PBP.  (Severity: 2)
Postfix control "unless" used at line 60, column 4.  See pages 96,97 of PBP.  (Severity: 2)
Link L<Bio::DB::SeqFeature> on line 76 does not specify text at line 72, column 1.  Without tex
Postfix control "unless" used at line 87, column 7.  See pages 96,97 of PBP.  (Severity: 2)
-1 is not one of the allowed literal values (0, 1, 2). Use the Readonly or Const::Fast module o
Ambiguously named variable "last" at line 113, column 3.  See page 48 of PBP.  (Severity: 3)
C-style "for" loop used at line 115, column 7.  See page 100 of PBP.  (Severity: 2)
Ambiguously named variable "last" at line 153, column 3.  See page 48 of PBP.  (Severity: 3)
C-style "for" loop used at line 157, column 7.  See page 100 of PBP.  (Severity: 2)
Postfix control "foreach" used at line 166, column 32.  See page 96 of PBP.  (Severity: 2)
Postfix control "foreach" used at line 241, column 18.  See page 96 of PBP.  (Severity: 2)
Postfix control "foreach" used at line 271, column 18.  See page 96 of PBP.  (Severity: 2)
Postfix control "foreach" used at line 288, column 18.  See page 96 of PBP.  (Severity: 2)
Subroutine name is a homonym for builtin function length at line 336, column 1.  See page 177 o
```

```
richards:biostudio-dev richards$ perlcritic --cruel --verbose '%p\n' lib/Bio/BioStu
Documentation::RequirePodSections
Documentation::RequirePodSections
Documentation::RequirePodSections
Documentation::RequirePodSections
Documentation::RequirePodSections
Documentation::RequirePodSections
Documentation::RequirePodSections
Document
ControlS
ControlS
ControlS
Document
ControlS
ValuesAn
NamingCo
ControlS
NamingCo
ControlS
ControlS
ControlS
ControlS
Subrouti
richards:biostudio-dev richards$
```

genedesign

GD Build | GD Run | CQC Build | CQC

```
GNU nano 2.0.6                                File: .perlcr

# perlcriticrc
[-Subroutines::ProhibitUnusedPrivateSubroutines]
[-Subroutines::ProhibitExcessComplexity]
[-Subroutines::ProhibitManyArgs]
[-ControlStructures::ProhibitPostfixControls]
```

# get critical

Upload files to be criticized
http://perlcritic.com/

Perl Best Practices on Safari Books Online (LBNL network only)
http://proquest.safaribooksonline.com/book/programming/perl/0596001738

Perl::Critic on CPAN, all the documenation you could want
http://search.cpan.org/~thaljef/Perl-Critic-1.118/lib/Perl/Critic.pm

how severe are your stylistic transgressions?

**perlcritic my_code.pl**

# Packaging Perl

sigils looks better
with a bow on

Repositories — bash — 95×32

GD Build | GD Run | CQC Build | CQC Run | JGISB Build

```
richards:Repositories richards$ module-starter --module=JGI::SynBio --author="S
 --email=SMRichardson@lbl.gov --builder='Module::Build'
Added to MANIFEST: Build.PL
Added to MANIFEST: Changes
Added to MANIFEST: ignore.txt
Added to MANIFEST: lib/JGI/SynBio.pm
Added to MANIFEST: MANIFEST
Added to MANIFEST: README
Added to MANIFEST: t/00-load.t
Added to MANIFEST: t/boilerplate.t
Added to MANIFEST: t/manifest.t
Added to MANIFEST: t/pod-coverage.t
Added to MANIFEST: t/pod.t
Created starter directories and files
richards:Repositories richards$
```

GeneDesign.pm | Build.PL

```perl
use 5.006;

use strict;
use warnings;

use Module::Build;
use File::Basename;
use Carp;

print "\n";

my $llt = 0;
my ($cpath,  $spath,  $tpath) =  (       q{},                  q{},        q{
my ($dcpath, $dspath, $dtpath) = ('/etc/GeneDesign/', '/usr/local/bin', '/tm
my ($g,  $e,  $bl,  $v) =  ( 0,   0,   0,   0);
my ($dg, $de, $dbl, $dv) = ("Y", "Y", "Y", "Y");

my $check = eval
{
  require Bio::GeneDesign::ConfigData;
};
if ($check)
{...}

my $GDB = Module::Build->new
(
    module_name         => 'Bio::GeneDesign',
    license             => 'bsd',
    dist_author         => q{Sarah Richardson <SMRichardson@lbl.gov>},
    dist_version        => '5.00',
    dist_abstract       => 'Functions for the design of synthetic genes',
    add_to_cleanup      => [ 'Bio::GeneDesign-*' ],
    create_makefile_pl  => 'traditional',
    build_requires      =>
    {
        'Test::More'      => 0.8,
        'Test::Deep'      => 0.1,
    },
    requires            =>
    {
        'perl'                => 5.006,
        'List::Util'          => 1.22,
        'Math::Combinatorics' => 0.09,
        'Text::Wrap'          => 2009.0305,
        'Bio::Root::Root'     => 0
    },
```

genedesign-dev

- bin
  - GD_Design_Building_Blocks_JGI.pl
  - GD_Design_Building_Blocks_JHU.pl
  - GD_Design_Building_Blocks.pl
  - GD_Design_Oligos_JGI.pl
  - GD_Design_Oligos_JHU.pl
  - GD_Filter_Enzymes.pl
  - GD_Generate_RSCU_Table.pl
  - GD_Graph_RSCU_Values.pl
  - GD_Juggle_Codons.pl
  - GD_Reverse_Translate.pl
  - GD_Sequence_Subtraction.pl
- BioGrepPatch
- codon_tables
- enzymes
- lib
  - Bio
    - GeneDesign
      - BuildingBlocks
      - Oligos
      - Basic.pm
      - Blast.pm
      - CodonJuggle.pm
      - Codons.pm
      - Exceptions.pm
      - Graph.pm
      - Oligo.pm
      - Palindrome.pm
      - Random.pm
      - RestrictionEnzyme.pm
      - RestrictionEnzymes.pm
      - ReverseTranslate.pm
      - SufTree.pm
      - Vector.pm
      - Vmatch.pm
    - GeneDesign.pm
- stubs

05-enzymes.t — genedesign-dev (git: master)

GeneDesign.pm | Build.PL | 04-codon_juggle.t | 05-enzymes.t

```
#! /usr/bin/perl -T

use Test::More tests => 2;

use Bio::GeneDesign;
use Bio::Seq;

use strict;
use warnings;

my $GD = Bio::GeneDesign->new();
$GD->set_organism(-organism_name => "yeast",
                  -table_path => "codon_tables/Standard.ct",
                  -rscu_path => "codon_tables/yeast.rscu");

$GD->set_restriction_enzymes(-list_path => "enzymes/test");

my $orf = "ATGGACAGATCTTGGAAGCAGAAGCTGAACCGCGACACCGTGAAGCTGACCGAGGTGATGACCTC
$orf .= "GAAGACCCGCCGCTAAATGGTTTTATACTTTAATTAATGCTAATTATTTGCCACCATGCCCACCCGA
$orf .= "ACCAAGATCACCGGCAGCAACAACTACCTGAGCCTGATCAGCCTGAACATCAACGGCCTGAACAGCC
$orf .= "ATCAAGCGGCACCGCCTGACCGACTGGCTGCACAAGCAGGACCCCACCTTCTGTTGCCTCCAGGAGA
$orf .= "CACCTGCGCGAGAAGGACCGGCACTACCTGCGGGTGAAGGGCTGGAAGACCATCTTTCAGGCCAACC
$orf .= "CTGAAGAAGCAGGCTGGCGTGGCCATCCTGATCAGCGACAAGATCGACTTCCAGCCCAAGGTGATCA
$orf .= "AAGGACAAGGAGGGCCACTTCATCCTGATCAAGGGCAAGATCCTGCAGGAGGAGCTGAGCATTCTGA
$orf .= "ATCTACGCCCCCAACGCCCGCGCCGCCACCTTCATCAAGGACACCCTCGTGAAGCTGAAGGCCCACA
$orf .= "GCTCCCCACACCATCATCGTCGGCGACCTGAACACCCCCCTGAGCAGTGA";
my $seqobj = Bio::Seq->new( -seq => $orf, -id => "torf");

#TESTING define_sites
my $rRES = {...};

my $tRES = $GD->enzyme_set;
is_deeply( $tRES, $rRES, "define_sites()" );

#TESTING define_site_status
my @enzes = values %$tRES;
my $rSITE_STATUS = {...};
my $tSITE_STATUS = $GD->restriction_status($seqobj);
is_deeply( $tSITE_STATUS, $rSITE_STATUS, "define_site_status()");
```

genedesign-dev

- Palindrome.pm
- Random.pm
- RestrictionEnzyme.pm
- RestrictionEnzymes.pm
- ReverseTranslate.pm
- SufTree.pm
- Vector.pm
- Vmatch.pm
- GeneDesign.pm
- stubs
- t
  - 00-load.t
  - 01-basic.t
  - 02-codons.t
  - 03-reverse_translate.t
  - 04-codon_juggle.t
  - 05-enzymes.t
  - 06-subtract_sequence.t
  - 08-graph.t
  - 09-palindrome.t
  - 10-vmatch.t
  - 11-blast.t
  - 12-suffix.t
  - 13-random.t
  - 14-scripts.t
  - boilerplate.t
  - manifest.t
  - pod-coverage.t
  - pod.t
  - testr_GRV_yeast.gif
- vectors
- Build
- Build.PL
- Changes
- MANIFEST
- META.json
- META.yml

GD Build | GD Run | CQC Build | CQC Run | JGISB Build | JGISB Run ... | BS Build | BS Run

```
richards:genedesign-dev richards$ sudo perl Build.PL

Where should configuration files be installed? [ /etc/GeneDesign/ ]
Where should scripts be installed? [ /usr/local/bin ]
Where should GeneDesign write tmp files? [ /tmp ]
Enable GD::Graphics support? [ Y ]
Enable EMBOSS palindrome for hairpin detection? [ Y ]
Enable BLAST+ for similarity detection? [ Y ]
Enable vmatch for similarity detection? [ Y ]
Configuration files will be in /etc/GeneDesign/
Scripts will be in /usr/local/bin
Will use BLAST+ executables found in /usr/local/bin/
Will use vmatch executables found in /usr/local/vmatch/

Created MYMETA.yml and MYMETA.json
Creating new 'Build' script for 'Bio-GeneDesign' version '5.00'

richards:genedesign-dev richards$ sudo ./Build && ./Build test
Building Bio-GeneDesign
t/00-load.t ............... ok
t/01-basic.t .............. ok
t/02-codons.t ............. ok
t/03-reverse_translate.t .. ok
t/04-codon_juggle.t ....... ok
t/05-enzymes.t ............ ok
t/06-subtract_sequence.t .. ok
t/08-graph.t .............. ok
t/09-palindrome.t ......... Warning: integer value out of range 20 more than (reset to) 17
t/09-palindrome.t ......... ok
t/10-vmatch.t ............ ok
t/11-blast.t .............. ok
t/12-suffix.t ............ ok
t/13-random.t ............ ok
t/boilerplate.t ........... ok
t/manifest.t .............. skipped: Author tests not required for installation
t/pod-coverage.t .......... ok
t/pod.t ................... ok
All tests successful.

Test Summary Report
-------------------
t/boilerplate.t         (Wstat: 0 Tests: 22 Failed: 0)
  TODO passed:   1-22
Files=17, Tests=131, 38 wallclock secs ( 0.08 usr  0.05 sys + 31.40 cusr  5.08 csys = 36.61 CPU)
Result: PASS
richards:genedesign-dev richards$ sudo ./Build install
Building Bio-GeneDesign
Installing /usr/local/share/man/man1/GD_Design_Building_Blocks.pl.1
Installing /usr/local/share/man/man1/GD_Design_Building_Blocks_JGI.pl.1
```

# Packaging Carrots

**version control:** manage releases and branches of an entire package, making it easy to refactor or upgrade multiple pieces of interacting code

**ninja deploys:** a cron job can automatically pull the latest version of your code from a repository, build it, test it, and deploy it on genepool while you eat lunch or nap! And all you had to do was tweak the defaults for a build script.

**portability:** packaging frameworks encourage you to think about your code running on systems other than your own, reporting to people who aren't you. This *will* make your code better.

**respectability:** packages look professional.

# wrap it up

The module starter documentation
http://search.cpan.org/~xsawyerx/Module-Starter-1.60/lib/Module/Starter.pm

The Module::Build documentation
http://search.cpan.org/~leont/Module-Build-0.4003/lib/Module/Build.pm

ExtUtils::MakeMaker if you prefer
http://search.cpan.org/~mschwern/ExtUtils-MakeMaker-6.64/lib/ExtUtils/MakeMaker.pm

give genepool a present

```
module-starter --module=JGI::MeMeME --author="Yours Truly"\
  --email=donotreply@gmail.com --builder= "Module::Build"
```

# Environment Modules

richards — bash — 52×21

```
richards:~ richards$ echo $PATH
/Users/richards/.rbenv/shims:/usr/local/bin:/Users/r
ichards/.rbenv/bin:/usr/local/sbin:/usr/local/mysql/
bin:/usr/local/vmatch:/usr/local/GeneDesign/bin:/usr
/local/BioStudio/bin:/usr/local/staden/bin:/usr/bin:
/bin:/usr/sbin:/sbin:/usr/local/bin:/opt/X11/bin:/us
r/X11/bin:/usr/local/ncbi/blast/bin:/usr/texbin
richards:~ richards$
```

**there has to be a better way**

**there is!**

# installing and running genedesign before environment modules

```
git pull origin master
perl Build.PL --install_base /jgi/tools/misc_bio/bioperl/bioperl-
live/lib/perl5
perl Deploy.PL --uselib "/jgi/tools/misc_bio/bioperl/bioperl-live/
lib/perl5 /jgi/tools/misc_bio/bioperl/bioperl-live/lib/perl5/
x86_64-linux-thread-multi" --binpath "/jgi/tools/groups/synbio/bin
--confpath "/jgi/tools/groups/synbio/GeneDesign" --bioperlbin "/
jgi/tools/misc_bio/bioperl/bioperl-live/bin"

/jgi/tools/groups/synbio/bin/GD_Design_Oligos.pl --help
```

# after environment modules

```
git pull origin master
perl Build.PL --like_last_time
./Build && ./Build test && ./Build install

module load perl EMBOSS vmatch blast+ genedesign

GD_Design_Oligos.pl --help
```