INCOMPLETE DRAFT: RNA –k-mers–> Protein

This manuscript (permalink) was automatically generated from czbiohub/kmerslay-paper@a4bbfd6 on March 17, 2020.

Authors

- **Olga Borisovna Botvinnik**
  0000-0003-4412-7970 · olgabot · olgabot
  Data Sciences Platform, Chan Zuckerberg Biohub

- **N. Tessa Pierce**
  0000-0002-2942-5331 · bluegenes
  Department of Population Health and Reproduction, University of California, Davis · Funded by NSF 1711984

Abstract

RNA-sequencing is a widely used measure of cellular state, and it is often used as proxy for estimating abundance of protein-coding molecules. However, the tools to estimate protein-coding sequence from RNA-seq reads don't allow for extraction of protein-coding sequences from distantly related species. We present `kmerslay`, a suite of k-mer based tools to extract protein-coding sequences from RNA-seq reads using reduced amino acid alphabets to allow for flexibility in proteome evolution. `kmerslay` can also perform all-by-all k-mer similarity of sequences in both amino acid and nucleotide reduced alphabets to find clusters of similar sequences which can then be used for homology inference. By enabling extraction of protein-coding sequences across a wide variety of species, `kmerslay` spearheads analysis of non-model organisms and their contribution to the evolution of life.

Introduction

Comparative transcriptomics, the study of gene expression profiles across species, provides a powerful view into the inner workings of cells, shared across millions of years of evolution. Homology assignment, the process of identifying genes conserved within evolution, across species continues to involve the requirement for an assembled and annotated genome in the species of interest. Unfortunately, 99.999% of the predicted 8.7 million Eukaryotic species on Earth have no submitted genome assembly [1,2], precluding the analysis of the vast majority of species on this planet. Adding to the difficulty, there is no consensus on the "best" way to assign homology, as each researcher may have different criteria for success, as demonstrated by the Quest for Orthologs consortium [3,4,5,6,7]. A promising, genome-agnostic method of assigning homology using reduced amino acid alphabets [8] has shown success in quickly finding similar protein-coding sequences. Thus, comparative transcriptomics remains hindered due to lack of complete genomes and the unsolved homology assignment problem.

To avoid genome assembly and homology assignment, we present `kmerslay`, a method to extract putative protein-coding regions from RNA-seq reads, using reduced amino acid alphabets. `kmerslay` provides several subcommands for working with RNA or protein sequences. To predict protein-coding sequence, (1) `kmerslay extract-coding` performs six-frame translation on RNA-seq short reads and returns peptide sequences with higher than expected by chance matches to a known index, internally using (2) `kmerslay bloom-filter` to create an index of known protein-coding sequences. To compute k-mer similarities across sequences, (3) `kmerslay compare-kmer-content` k-merizes each sequence by running a sliding window of size k along the sequence, [illegible] word, and computing the number of overlapping k-long sequences shared across pairs of sequences. By sidestepping genome assembly, gene annotation, and homology assignment, `kmerslay` empowers non-model organism researchers to investigate their taxa of interest without fear.

Implementation

Reduced alphabets

At the core of `kmerslay` is the ability to cheaply compare sequences using k-mers. As k-mers are very brittle to substitutions and thus to compare across species, one must allow for minor base substitutions that still maintain similar chemical or structural properties. A reduced alphabet can encode useful information into a smaller alphabet space, and enable sequence comparisons across a broader variety of species than the original alphabet alone.

Reduced amino acid alphabets

Reduced amino acid alphabets have been useful for over 50 years [9] in finding related protein sequences [10,11,12,13,14]. Recently, a reduced amino acid alphabet (specifically, `aa9` below) combined with k-mers were used to find homologous protein-coding sequences [8]. We build on this concept by enabling prediction of protein-coding sequences from RNA-seq reads, and by enabling users to perform a parameter sweep in an all-by-all comparison to identify putative homologs using a variety of alphabet metrics.

Dayhoff and `HP` alphabets

Table 1: Dayhoff and hydrophobic-polar encodings are a reduced amino acid alphabet allowing for permissive cross-species sequence comparisons. For example, the amino acid sequence `SASHAFIERCE` would be Dayhoff-encoded to `bbbdbfecdac`, and HP-encoded to `phpphhhpppp`, as below.

Amino acid	Property	Dayhoff	Hydrophobic-polar (HP)
`C`	Sulfur polymerization	`a`	`p`
`A`, `G`, `P`, `S`, `T`	Small	`b`	`AGP`: h `ST`: `p`
`D`, `E`, `N`, `Q`	Acid and amide	`c`	`p`
`H`, `K`, `R`	Basic	`d`	`p`
`I`, `L`, `M`, `V`	Hydrophobic	`e`	`h`
`F`, `W`, `Y`	Aromatic	`f`	`h`

```
protein20: SASHAFIERCE
dayhoff6:  bbbdbfecdac
hp2:       phpphhhpppp
```

All implemented alphabets (with citations, not as nicely organized)

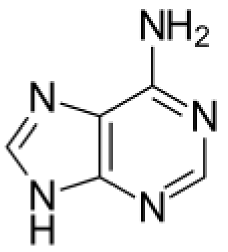
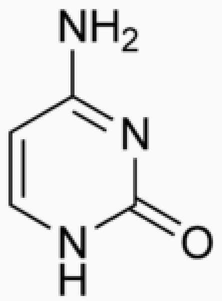
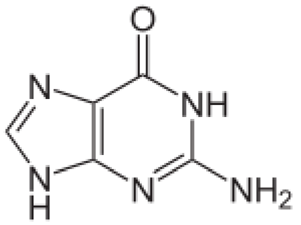
[NOTE: maybe this should go into the supplementary? The main alphabets that have been successful for me are dayhoff and HP]

Citation	Alphabet	Amino acid groups
Typesetting math: 100% 15]	hp2	`AFGILMPVWY` `CDEHKNQRST`

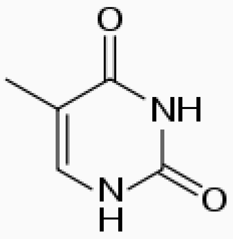
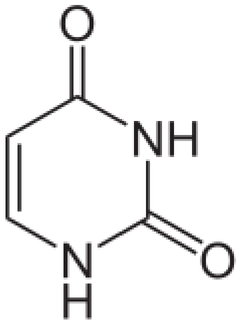
Citation	Alphabet	Amino acid groups
Peterson, E. L., *et al.* (2009) [10]	gbmr4	`G` `ADKERNTSQ` `YFLIVMCWH` `P`
Dayhoff, M. O., & Eck, R. V. (1968). [9]	dayhoff6	`AGPST` `HRK` `DENQ` `FWY` `ILMV` `C`
This paper	botvinnik8	`AG` `DE` `RK` `NQ` `ST` `FY` `LIV` `CMWHP`
Hu, X., & Friedberg, I. (2019). [8]	aa9	`G` `AST` `KR` `EQ` `DN` `CFILMVY` `W` `H` `P`
Peterson, E. L., *et al.* (2009) [10]	sdm12	`G` `A` `D` `KER` `N` `TSQ` `YF` `LIVM` `C` `W` `H` `P`
Peterson, E. L., *et al.* (2009) [10]	hsdm17	`G` `A` `D` `KE` `R` `N` `T` `S` `Q` `Y` `F` `LIV` `M` `C` `W` `H` `P`
Dayhoff, M. O., & Eck, R. V. (1968). [9]	protein20	`G` `A` `D` `E` `K` `R` `N` `T` `S` `Q` `Y` `F` `L` `I` `V` `M` `C` `W` `H` `P`

Reduced nucleotide alphabets

The IUPAC degenerate nucleotide code [16] includes several two-letter codes for the original 4-letter nucleobase alphabet. The first, Weak/Strong, indicates the strength of the hydrogen bond across the double strand. The bond of adenine to thymine has two hydrogen bonds, making it weak; and the bond of guanine to cytosine has three hydrogen bonds, making it 50% stronger. The second, Purine/Pyrimidine, encodes the ring size of the nucleobase, where Adenine and Guanine both have larger Purine double rings, while Cytosine and Thymine/Uracil have smaller Pyrimidine rings. The third, Amino/Keto, designates the main functional group of the ring, where Adenine and Cytosine both have an Amino group, while Guanine and Thymine/Uracil both have a Keto group.

Nucleotide	Hydrogen Bonding	Ring type	Ring functional group	Nucleobase chemical structure
A	Weak (W)	Purine (R)	Amino (M)	
C	Strong (S)	Pyrimidine (Y)	Amino (M)	
G	Strong (S)	Purine (R)	Keto (K)	

Typesetting math: 100%

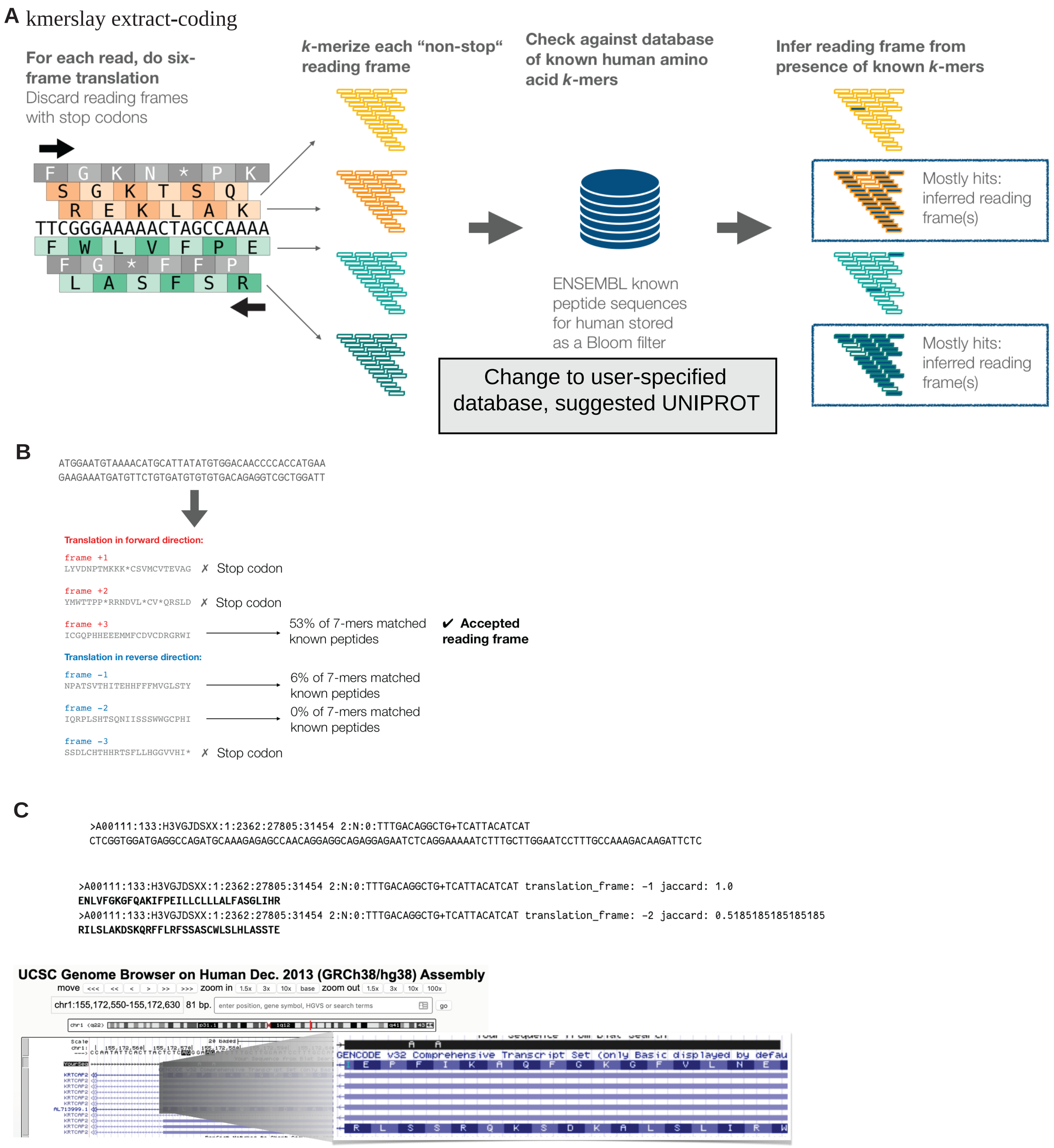
Nucleotide	Hydrogen Bonding	Ring type	Ring functional group	Nucleobase chemical structure
T	Weak (W)	Pyrimidine (Y)	Keto (K)	
U	Weak (W)	Pyrimidine (Y)	Keto (K)	

Thus, the nucleotide string `GATTACA` would be re-encoded into the following:

```
Nucleotide:         GATTACA
Weak/Strong:        SWWWWSW
Purine/Pyrimidine:  RRYYRYR
Functional group:   KMKKMMM
```

`kmerslay extract-coding`

Typesetting math: 100%


A
kmerslay extract-coding
For each read, do six-frame translation
Discard reading frames with stop codons
F G K N * P K
S G K T S Q
R E K L A K
TTCGGGAAAAACTAGCCAAAA
F W L V F P E
F G * F F P
L A S F S R
k-merize each "non-stop" reading frame
Check against database of known human amino acid k-mers
ENSEMBL known peptide sequences for human stored as a Bloom filter
Change to user-specified database, suggested UNIPROT
Infer reading frame from presence of known k-mers
Mostly hits: inferred reading frame(s)
Mostly hits: inferred reading frame(s)
B
Translation in forward direction:
✗ Stop codon
✗ Stop codon
53% of 7-mers matched known peptides
✔ Accepted reading frame
Translation in reverse direction:
6% of 7-mers matched known peptides
0% of 7-mers matched known peptides
✗ Stop codon
C
UCSC Genome Browser on Human Dec. 2013 (GRCh38/hg38) Assembly
move <<< << < > >> >>> zoom in 1.5x 3x 10x base zoom out 1.5x 3x 10x 100x
chr1:155,172,550-155,172,630 81 bp.
enter position, gene symbol, HGVS or search terms
go
chr1 (q22)
p31.1
1q12
q41
43
44
GENCODE v32 Comprehensive Transcript Set (only Basic displayed by defau
E P F I K A Q F G K G F V L N E
R L S S R Q K S D K A L S L I R W
KRTCAP2
AL713999.1

```
ATGGAATGTAAAACATGCATTATATGTGGACAACCCCACCATGAA
GAAGAAATGATGTTCTGTGATGTGTGTGACAGAGGTCGCTGGATT
```

```
frame +1
LYVDNPTMKKK*CSVMCVTEVAG

frame +2
YMWTTPP*RRNDVL*CV*QRSLD

frame +3
ICGQPHHEEEMMFCDVCDRGRWI
```

```
frame -1
NPATSVTHITEHHFFFMVGLSTY

frame -2
IQRPLSHTSQNIISSSWWGCPHI

frame -3
SSDLCHTHHRTSFLLHGGVVHI*
```

```
>A00111:133:H3VGJDSXX:1:2362:27805:31454 2:N:0:TTTGACAGGCTG+TCATTACATCAT
CTCGGTGGATGAGGCCAGATGCAAAGAGAGCCAACAGGAGGCAGAGGAGAATCTCAGGAAAAATCTTTGCTTGGAATCCTTTGCCAAAGACAAGATTCTC
```

```
>A00111:133:H3VGJDSXX:1:2362:27805:31454 2:N:0:TTTGACAGGCTG+TCATTACATCAT translation_frame: -1 jaccard: 1.0
ENLVFGKGFQAKIFPEILLCLLLALFASGLIHR
>A00111:133:H3VGJDSXX:1:2362:27805:31454 2:N:0:TTTGACAGGCTG+TCATTACATCAT translation_frame: -2 jaccard: 0.5185185185185185
RILSLAKDSKQRFFLRFSSASCWLSLHLASSTE
```

Overview of `kmerslay extract-coding` **A.** First, each read is translated into all six possible protein-coding translation frames. Next, reading frames with stop codons are eliminated. Each protein-coding frame is k -merized, then the fraction of k-mers which appear in the known protein-coding database is computed. Frames which contain a fraction of coding frames exceeding the threshold are inferred to be putatively protein-coding. **B.** Worked example of an RNA-seq read with a single putatitive reading frame. **C.** Worked example of an RNA-seq read with multiple reading frames, and a `UCSC` genome browser shot of the read showing that both reading frames are present in the annotation.

Typesetting math: 100%

Set Jaccard threshold of `extract-coding` by controlling false positive rate of protein-coding prediction

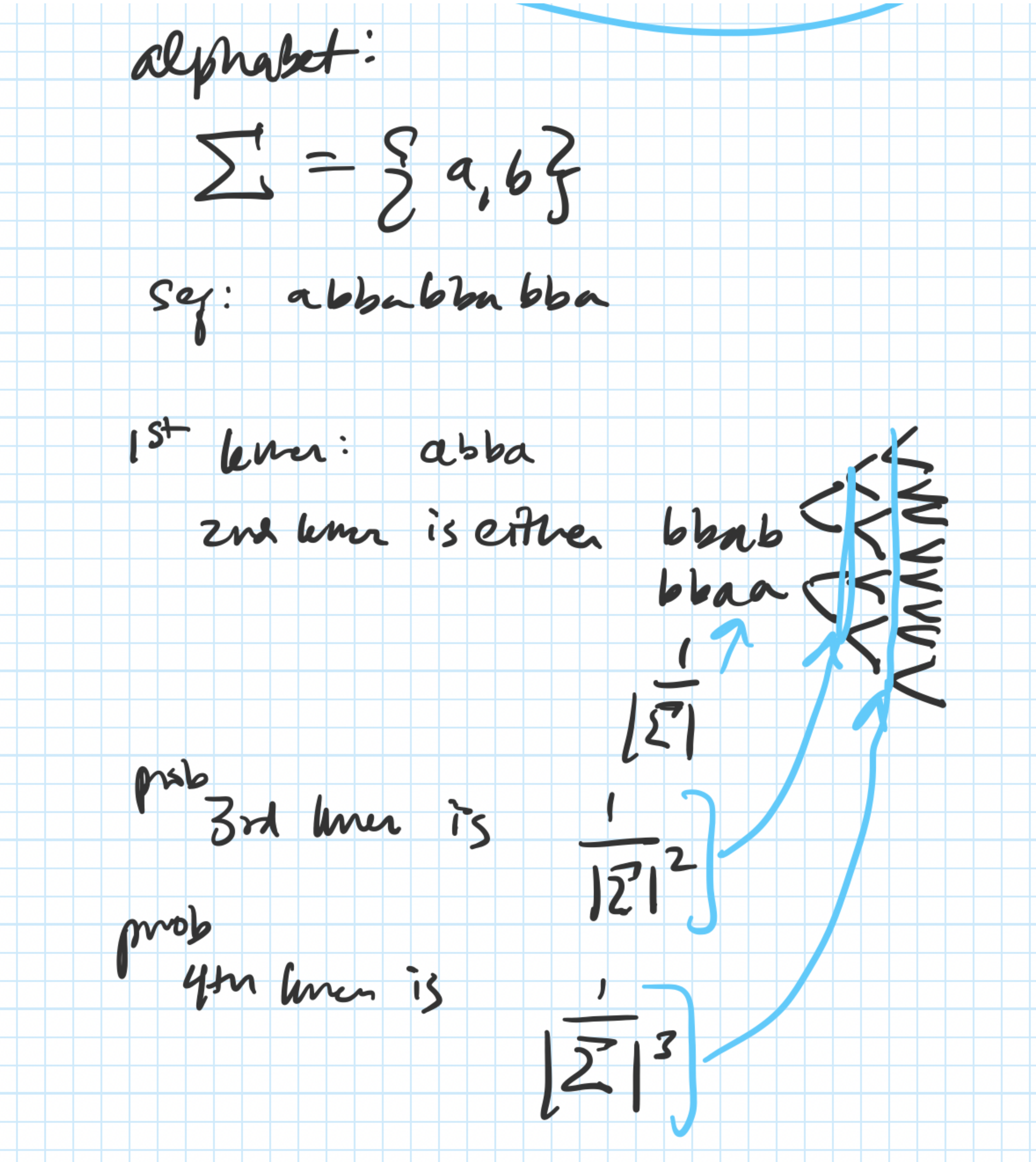
To set a threshold of the minimum Jaccard overlap between a translated read's frame and the reference proteome, the most statistically principled way is to control the false positive rate of predicing a protein-coding read.

Probability of random k-mers from a read

If k-mers from reads were independent, identically distributed (iid) variables, then a translated read of length $L_{\mathrm{translated}}$ drawing letters from the alphabet Σ, whose size is $\left|\Sigma\right|$, would contain

\[\left(\frac{1}{\left| \Sigma \right|^k \right)^{L_{\mathrm{translated}} - k + 1} \] **(1)**

However, k-mers drawn from reads are not iid. Let's take a simple example. If we have a two-letter alphabet, $\Sigma = a, b,$, thus $\left|\Sigma\right| = 2$. Let us use an example sequence $S = abbabba$. If $k = 4$, then the first k-mer is $abba$. The second k-mer is thus either $bbaa$ or $bbab$, with equal probability. We can generalize this: Given the first k-mer, the first $k - 1$ letters from the second k-mer are known, and thus the probability of guessing the next k-mer is $\frac{1}{\left|\Sigma\right|}$.



Probability of future k -mers is influenced by the existence of previous k-mers.

Thus, the probability of a random Probability of a random k-mer from a sequencing read is completely dependent on the alphabet size $|\Sigma|$ and its translated sequence length, $L_{\mathrm{translated}}$:

\[\mathrm{Pr}(\mathrm{random } k-\mathrm{-mer in protein translation}) &= \frac{1}{\left| \Sigma \right|^k * \left(\frac{1}{\left| \Sigma \right| \right)^{L_{\mathrm{translated}} - k + 1}\\ &= \frac{1}{\left| \Sigma \right|^k * \left| \Sigma \right| \right)^{L_{\mathrm{translated}} - k + 1}} \\ &= \frac{1}{\left| \Sigma \right|^L_{\mathrm{translated}}}\] (2)

Bloom filter collision probability

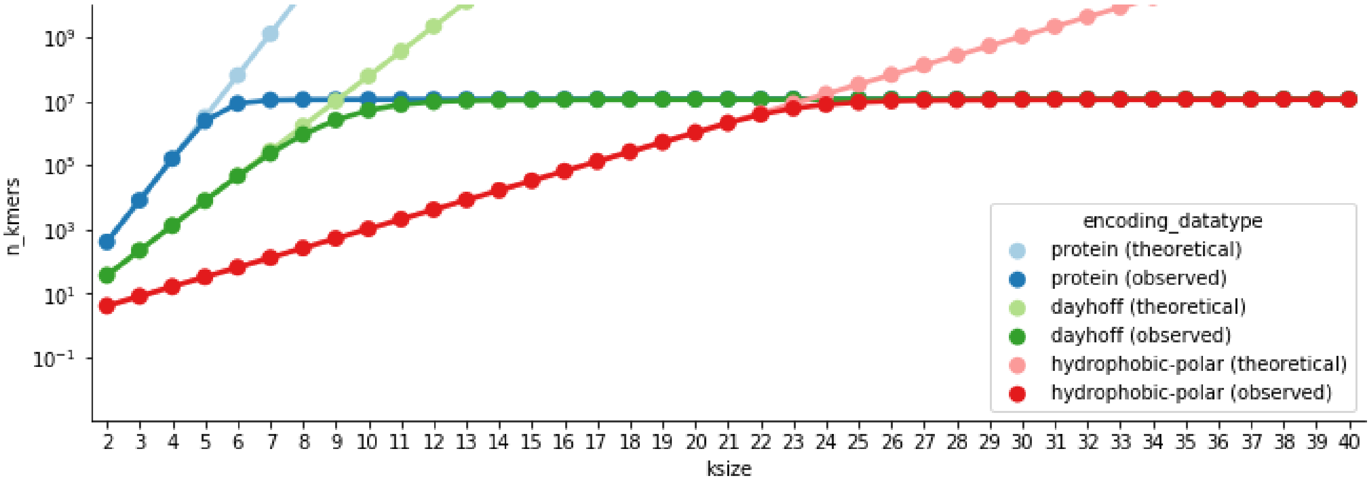
Typesetting math: 100%

The probability of error of the `khmer` bloom filter implementation [17] used in `kmerslay`, given N distinct k-mers counted, a hash table size of H, and Z total number of hash tables, is

$$\mathrm{Pr}(\mathrm{falsepositiveinbloomfilter}) = \left(1 - \exp^{N/H}\right)^{Z}. \tag{3}$$

Theoretically, the total number of k-mers is limited by the alphabet size and choice of k. Empirically, the number of possible k-mers is limited by the k-mers which are compatible with life, and by $k = 5$, the number of theoretical protein k-mers exceeds the number of observed protein k-mers. Additionally, the mass of all possible k-mers of a certain size, exceeds the mass of the planet earth by $k = X$ [get the data for this]. The UniProtKB Opisthokonta manually reviewed dataset contains 4.8×10^7 7-mers in the protein alphabet. Thus, we can give an upper bound to the number of theoretical k-mers to be 10^8. Therefore, the total number of k-mers in the bloom filter is,

$$N = \min\left(10^8, |\Sigma|^k\right). \tag{4}$$



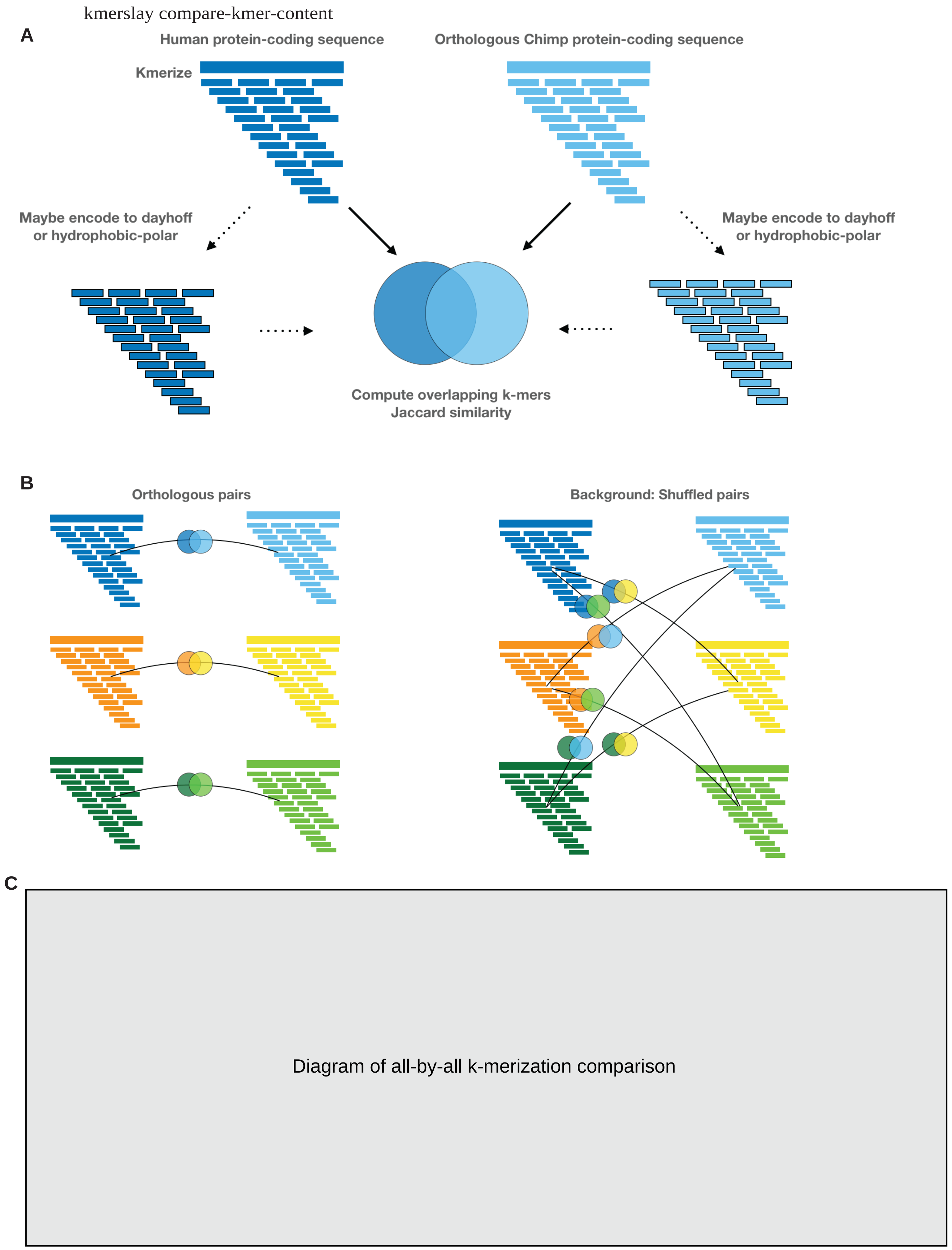
Number of theoretical k -mers given alphabet size, compared to observed k-mers in ENSEMBL human translated proteome. The number of observed k-mers is distinct from the number of theoretical k-mers, as the total number of observed k-mers is limted by k-mers compatible with life. Rerun this with uniprot uniref data.

False positive rate of protein-coding prediction

Combining Equations 2, 4, and 3, for an RNA-seq read of length L where its translated length $L_{\mathrm{translated}} = \lfloor \frac{L}{3} \rfloor$, containing a possible six frames of translation, then

[Math Processing Error] (5)

`kmerslay compare-kmer-content` performs all-by-all or pairwise k-mer similarity of protein or nucleotide sequences using reduced alphabets


kmerslay compare-kmer-content
A
Human protein-coding sequence
Orthologous Chimp protein-coding sequence
Kmerize
Maybe encode to dayhoff or hydrophobic-polar
Maybe encode to dayhoff or hydrophobic-polar
Compute overlapping k-mers
Jaccard similarity
B
Orthologous pairs
Background: Shuffled pairs
C
Diagram of all-by-all k-merization comparison

Overview of `kmerslay compare-kmer-content` **A.** Protein sequences are k -merized by converting into a bag of words using a sliding window of size k, potentially re-encoded to a lossy alphabet, and then their fraction of overlapping k-mers is computed into a Jaccard similarity. **B.** One option for `kmerslay compare-kmer-content` is to specify a pair of sequence files, and compute a background of k-mer similarty using randomly shuffled pairs. **C.** Another option for `kmerslay compare-kmer-content` is to do an all-by-all k-mer similarity comparison.

Applications

Installation

`kmerslay` can be installed with the Anaconda package manager, `conda` (preferred),

```
# Note: not actually on bioconda yet ... this is aspirational
conda install --channel bioconda kmerslay
```

or from the Python Package Index (PyPI) with the Python package manager, `pip`,

```
# Note: not actually on PyPI yet ... this is aspirational
pip install kmerslay
```

Prediction of protein-coding sequences across a variety of species

We used `kmerslay extract-coding` to obtain putative protein-coding sequences from a comaparative transcriptomic dataset spanning nine species and six tissues [18].

Read preprocessing

As the protein-coding score is assessed on the entire read, we recommend RNA-seq reads be removed of library artifacts to the best of the user's ability. This means, the adapters should be trimmed, and if there was a negative insert size such that the R1 and R2 reads overlap, then the read pairs should be merged.

Creation of amino acid k-mer database with `kmerslay bloom-filter`

Before predicting protein-coding sequences, `kmerslay` must create a database of known amino acid k-mers, which is stored in the form of a probabilistic set membership data structure known as a bloom filter. `kmerslay` uses the bloom filter implementation in `khmer`/`oxli` [17,19], called a NodeGraph. We created a dataset of known amino acid k-mers from the manually annotated UniProtKB/Swiss-Prot databases [20,21]. We used only protein sequences observed in *Opisthokont* species [22], previously known as a "Fungi/Metazoa" group that encompasseses "Fungus-like" *Holomycota* and "Animal-like" *Holozoa*. [NOTE: Does this need a figure/phylogenetic timetree?]

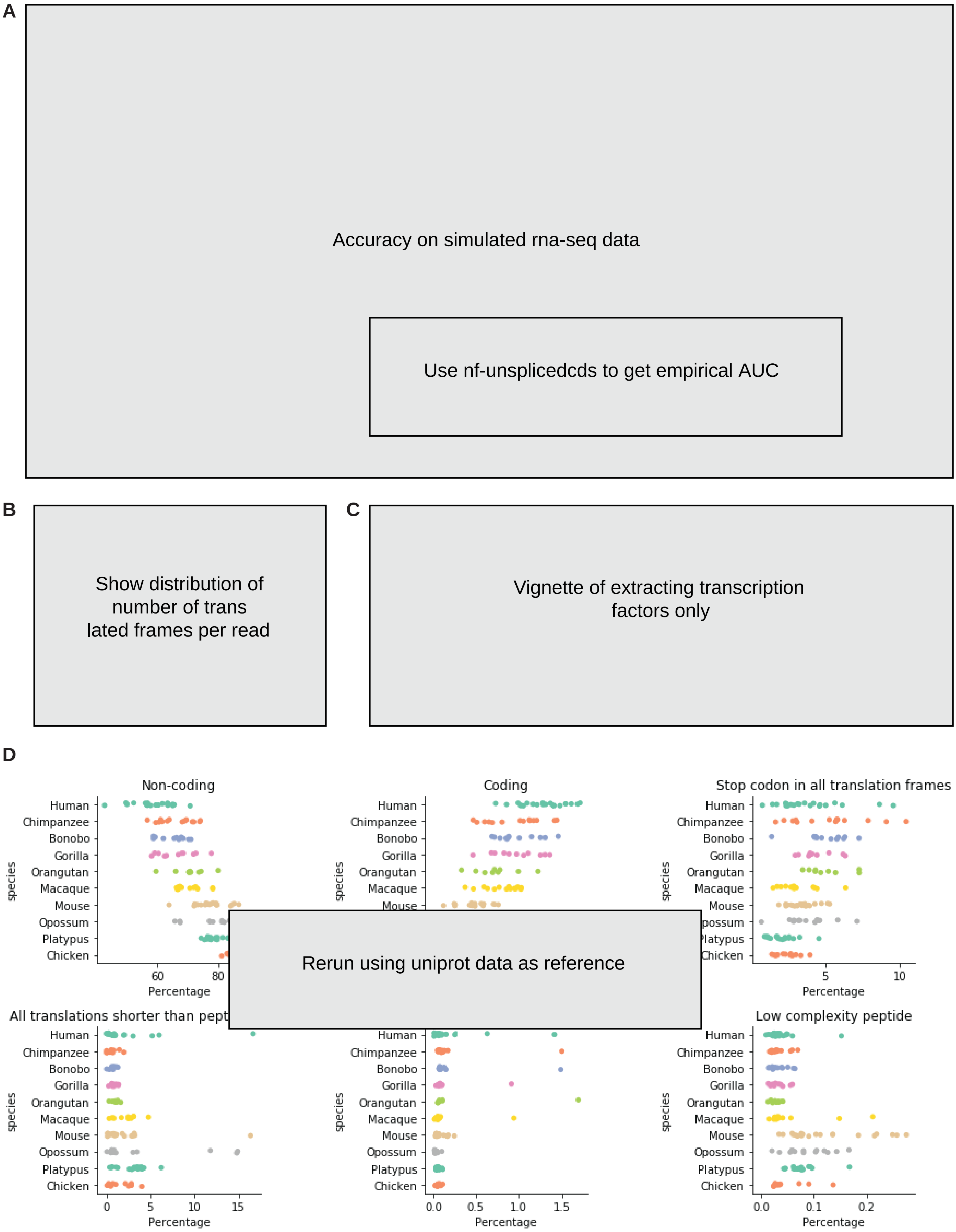
```
kmerslay bloom-filter \
  --tablesize 100000000 \
  --molecule protein \
  --peptide-ksize 7 \
  --save-as uniprot-reviewed_yes+taxonomy_2759__molecule-protein_ksize-
7.bloomfilter \
  uniprot-reviewed_yes+taxonomy_2759.fasta.gz
```

Prediction of protein-coding sequences with `kmerslay extract-coding`

We then predicted protein coding reads using the created bloom filter using `kmerslay extract-`

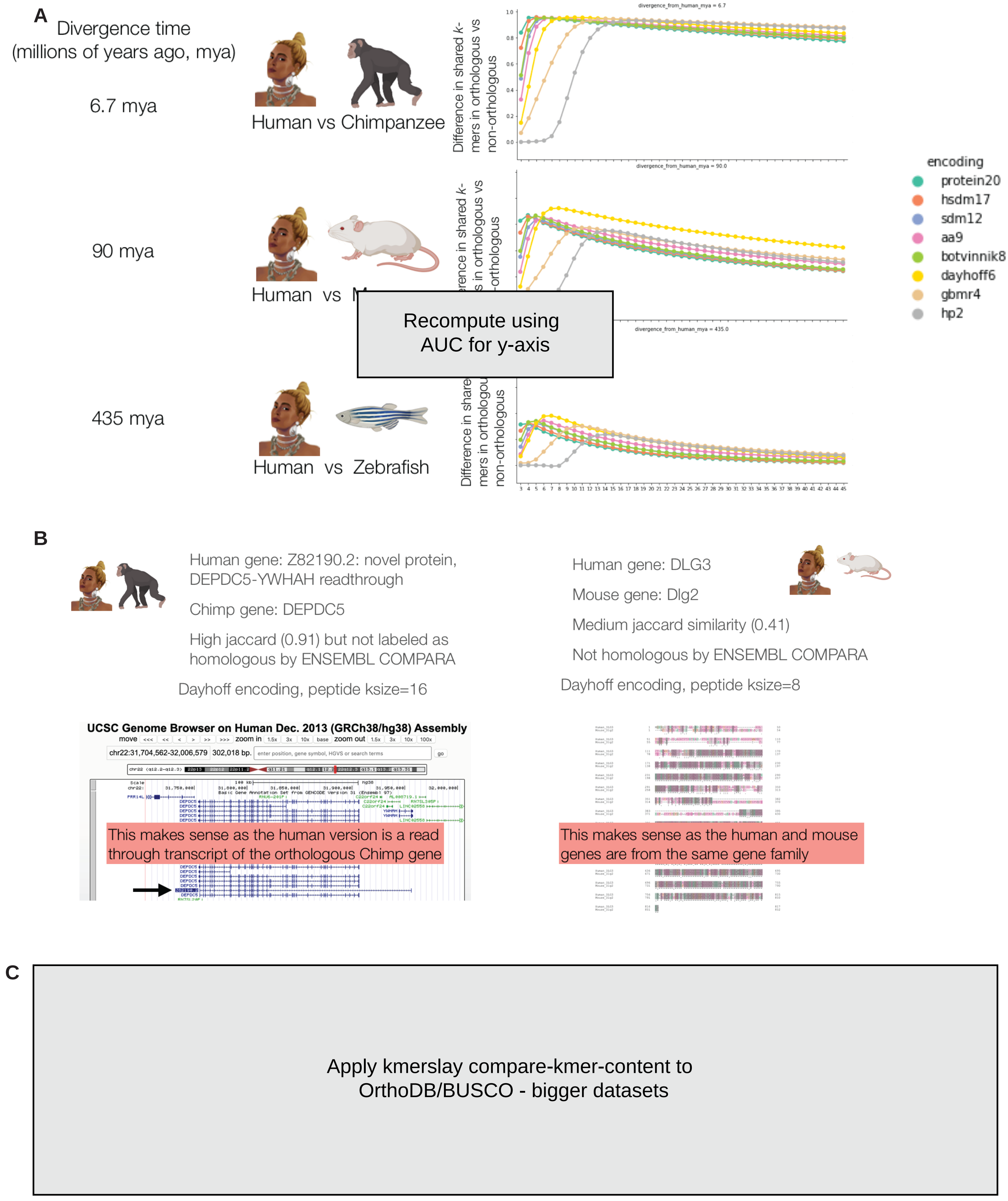
```
kmerslay extract-coding \
  --molecule protein \
  --coding-nucleotide-fasta
SRR306800_GSM752653_ggo_br_F_1__coding_reads_nucleotides.fasta \
  --csv SRR306800_GSM752653_ggo_br_F_1__coding_scores.csv \
  --json-summary SRR306800_GSM752653_ggo_br_F_1__coding_summary.json \
  --jaccard-threshold 0.8 \
  --peptides-are-bloom-filter \
  uniprot-reviewed_yes+taxonomy_2759__molecule-protein_ksize-7.bloomfilter \
  SRR306800_GSM752653_ggo_br_F_1_trimmed.fq.gz >
SRR306800_GSM752653_ggo_br_F_1__coding_reads_peptides.fasta
```

Typesetting math: 100%



Applications of `kmerslay extract-coding`. **A.** We simulated RNA-seq data using Opisthokonta species from the Quest for Orthologs dataset for true positive protein-coding RNAs, reads completely contained within intergenic, intronic, and UTR sequences as true positive noncoding RNAs, and reads partially overlapping a coding and noncoding region as an adversarial test set. We then predicted protein-coding sequences and computed false positive and false negative rates. False Positive coding reads were found to be … False negative noncoding reads were found to be … **B.** Number of putative protein-coding sequences per read. **C.** This method could also be used to extract only reads whose putative protein-coding sequences are transcription factors. **D.** We ran `kmerslay extract-coding` on the five tissues and nine species from the Brawand 2011 dataset.

`kmerslay compare-kmer-content` is a simple method to identify homologs



Recompute using AUC for y-axis

Human gene: Z82190.2: novel protein, DEPDC5-YWHAH readthrough

Chimp gene: DEPDC5

High jaccard (0.91) but not labeled as homologous by ENSEMBL COMPARA

Dayhoff encoding, peptide ksize=16

This makes sense as the human version is a read through transcript of the orthologous Chimp gene

Human gene: DLG3

Mouse gene: Dlg2

Medium jaccard similarity (0.41)

Not homologous by ENSEMBL COMPARA

Dayhoff encoding, peptide ksize=8

This makes sense as the human and mouse genes are from the same gene family

Apply kmerslay compare-kmer-content to OrthoDB/BUSCO - bigger datasets

Typesetting math: 100%

Applications of `kmerslay compare-kmer-content`. **A.** We used `kmerslay compare-kmer-content` on pairs of orthologous protein sequences between humans and the remaining Opisthokonta species in the Quest for Orthologs dataset. x-axis, k -mer size, y-axis, mean difference. **B.** False positive calls by `kmerslay compare-kmer-content` are either paralogs or read-through protein products. **C.** We applied `kmerslay compare-kmer-content` to … to find putative orthologs. We found … the accuracy was …

Discussion

Bold text

Semi-bold text

Centered text

Right-aligned text

Italic text

Combined *italics and **bold***

~~Strikethrough~~

1. Ordered list item
2. Ordered list item
    a. Sub-item
    b. Sub-item
        i. Sub-sub-item
3. Ordered list item
    a. Sub-item

- List item
- List item
- List item

subscript: H_2O is a liquid

superscript: 2^{10} is 1024.

unicode superscripts^{0123456789}

unicode subscripts$_{0123456789}$

A long paragraph of text. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Putting each sentence on its own line has numerous benefits with regard to editing and version control.

Line break without starting a new paragraph by putting
Typesetting math: 100% ne.

Document organization

Document section headings:

Heading 1

Heading 2

Heading 3

Heading 4

A heading centered on its own printed page

Typesetting math: 100%

Horizontal rule:

`Heading 1`'s are recommended to be reserved for the title of the manuscript.

`Heading 2`'s are recommended for broad sections such as *Abstract*, *Methods*, *Conclusion*, etc.

`Heading 3`'s and `Heading 4`'s are recommended for sub-sections.

Links

Bare URL link: https://manubot.org

Long link with lots of words and stuff and junk and bleep and blah and stuff and other stuff and more stuff yeah

Link with text

Link with hover text

Link by reference

Citations

Citation by DOI [23].

Citation by PubMed Central ID [24].

Citation by PubMed ID [25].

Citation by Wikidata ID [26].

Citation by ISBN [27].

Citation by URL [28].

Citation by tag [29].

Multiple citations can be put inside the same set of brackets [23,27,29]. Manubot plugins provide easier, more convenient visualization of and navigation between citations [24,25,29,30].

Citation tags (i.e. aliases) can be defined in their own paragraphs using Markdown's reference link syntax:

Referencing figures, tables, equations

Figure 1

Figure 2

Typesetting math: 100%

Figure 3

Figure 4

Table 1

Equation 1

Equation 2

Quotes and code

> Quoted text

> Quoted block of text
>
> Two roads diverged in a wood, and I—
> I took the one less traveled by,
> And that has made all the difference.

Code `in the middle` of normal text, aka `inline code`.

Code block with Python syntax highlighting:

```python
from manubot.cite.doi import expand_short_doi

def test_expand_short_doi():
    doi = expand_short_doi("10/c3bp")
    # a string too long to fit within page:
    assert doi == "10.25313/2524-2695-2018-3-vliyanie-enhansera-copia-i-
        insulyatora-gypsy-na-sintez-ernk-modifikatsii-hromatina-i-
        svyazyvanie-insulyatornyh-belkov-vtransfetsirovannyh-geneticheskih-
        konstruktsiyah"
```

Code block with no syntax highlighting:

```
Exporting HTML manuscript
Exporting DOCX manuscript
Exporting PDF manuscript
```

Figures

Typesetting math: 100%

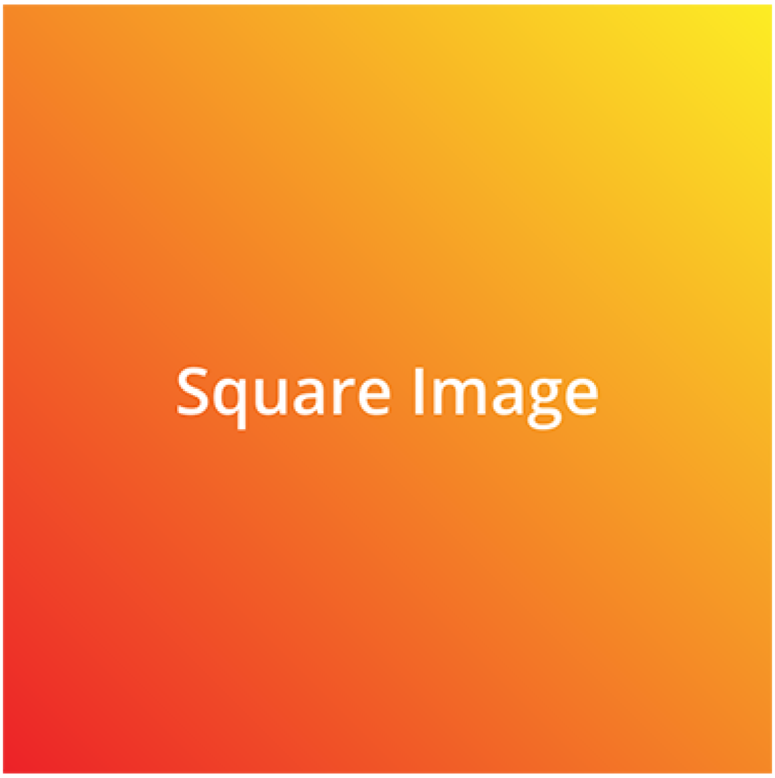


Figure 1: A square image at actual size and with a bottom caption. Loaded from the latest version of image on GitHub.

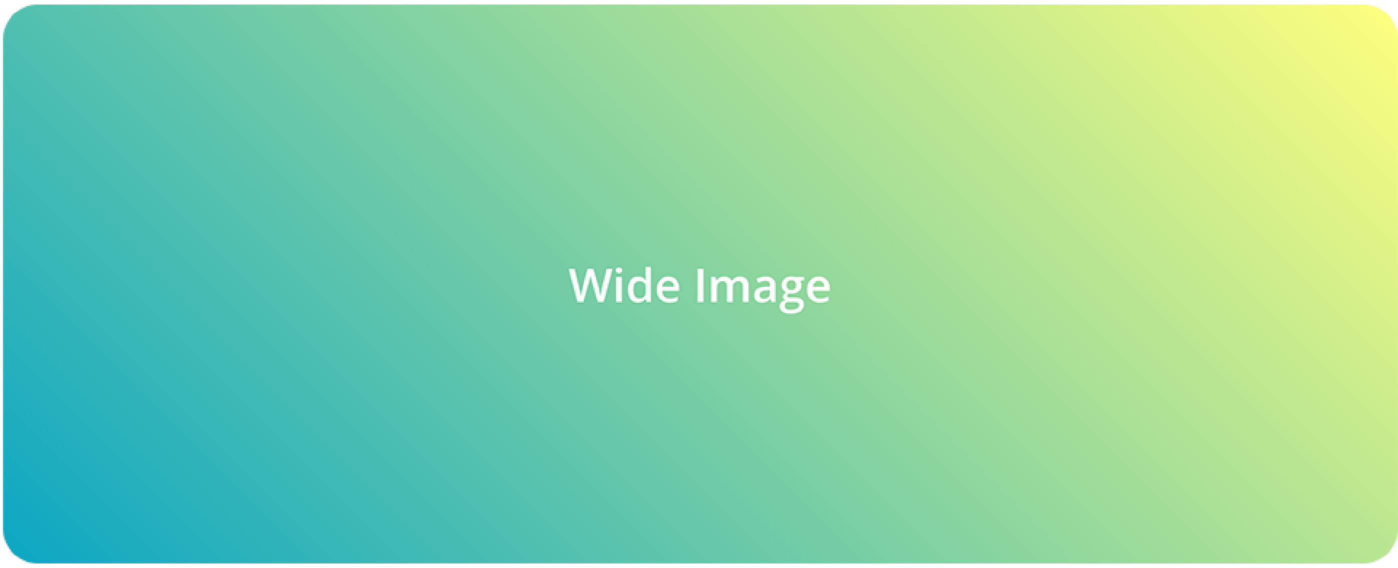


Figure 2: An image too wide to fit within page at full size. Loaded from a specific (hashed) version of the image on GitHub.

Typesetting math: 100%

Figure 3: A tall image with a specified height. Loaded from a specific (hashed) version of the image on GitHub.

Figure 4: A vector `.svg` image loaded from GitHub. The parameter `sanitize=true` is necessary to properly load SVGs hosted via GitHub URLs. White background specified to serve as a backdrop for transparent sections of the image.

Tables

Table 1: A table with a top caption and specified relative column widths.

Bowling Scores	**Jane**	**John**	**Alice**	**Bob**
Game 1	150	187	210	105
Game 2	98	202	197	102
Game 3	123	180	238	134

Table 2: A table too wide to fit within page.

	Digits 1-33	**Digits 34-66**	**Digits 67-99**	**Ref.**
pi	3.14159265358979323846264338327950	288419716939937510582097494459230	781640628620899862803482534211706	`piday.org`
e	2.71828182845904523536028747135266	249775724709369995957496696762772	407663035354759457138217852516642	`nasa.gov`

Typesetting math: 100% ged cells using the `attributes` plugin.

	Colors	
Size	**Text Color**	**Background Color**
big	blue	orange
small	black	white

Equations

A LaTeX equation:

$$\int_0^\infty e^{-x^2} dx = \frac{\sqrt{\pi}}{2} \tag{1}$$

An equation too long to fit within page:

$$\begin{split} x = a + b + c + d + e + f + g + h + i + j + k + l + m + n + o + p + q + r + s + t \\ + u + v + w + x + y + z + 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 \end{split} \tag{2}$$

Special

⚠ WARNING *The following features are only supported and intended for `.html` and `.pdf` exports. Journals are not likely to support them, and they may not display correctly when converted to other formats such as `.docx`.*

LINK STYLED AS A BUTTON

Adding arbitrary HTML attributes to an element using Pandoc's attribute syntax:

> Manubot Manubot Manubot Manubot Manubot. Manubot Manubot Manubot Manubot. Manubot Manubot Manubot. Manubot Manubot. Manubot.

Adding arbitrary HTML attributes to an element with the Manubot `attributes` plugin (more flexible than Pandoc's method in terms of which elements you can add attributes to):

> Manubot Manubot Manubot Manubot Manubot. Manubot Manubot Manubot Manubot. Manubot Manubot Manubot. Manubot Manubot. Manubot.

Available background colors for text, images, code, banners, etc:

`white` `lightgrey` `grey` `darkgrey` `black` `lightred` `lightyellow` `lightgreen` `lightblue` `lightpurple` `red` `orange` `yellow` `green` `blue` `purple`

Using the Font Awesome icon set:

✓ ? ★ 🔔 ⊗ ⋯

Typesetting math: 100%

Light Grey Banner
useful for *general information* - manubot.org

Blue Banner
useful for *important information* - manubot.org

Light Red Banner
useful for *warnings* - manubot.org

Supplementary

Typesetting math: 100%

References

1. **How Many Species Are There on Earth and in the Ocean?**
Camilo Mora, Derek P. Tittensor, Sina Adl, Alastair G. B. Simpson, Boris Worm
PLoS Biology (2011-08-23) https://doi.org/fpr4z8
DOI: 10.1371/journal.pbio.1001127 · PMID: 21886479 · PMCID: PMC3160336

2. **Genome List - Genome - NCBI** https://www.ncbi.nlm.nih.gov/genome/browse

3. **The origin and evolution of cell types**
Detlev Arendt, Jacob M. Musser, Clare V. H. Baker, Aviv Bergman, Connie Cepko, Douglas H. Erwin, Mihaela Pavlicev, Gerhard Schlosser, Stefanie Widder, Manfred D. Laubichler, Günter P. Wagner
Nature Reviews Genetics (2016-11-07) https://doi.org/f9b62x
DOI: 10.1038/nrg.2016.127 · PMID: 27818507

4. **Advances and Applications in the Quest for Orthologs**
Natasha Glover, Christophe Dessimoz, Ingo Ebersberger, Sofia K Forslund, Toni Gabaldón, Jaime Huerta-Cepas, Maria-Jesus Martin, Matthieu Muffato, Mateus Patricio, Cécile Pereira, … Paul D Thomas
Molecular Biology and Evolution (2019-10) https://doi.org/ggcncc
DOI: 10.1093/molbev/msz150 · PMID: 31241141 · PMCID: PMC6759064

5. **Gearing up to handle the mosaic nature of life in the quest for orthologs**
Kristoffer Forslund, Cecile Pereira, Salvador Capella-Gutierrez, Alan Sousa da Silva, Adrian Altenhoff, Jaime Huerta-Cepas, Matthieu Muffato, Mateus Patricio, Klaas Vandepoele, Ingo Ebersberger, … Quest for Orthologs Consortium
Bioinformatics (2018-01-15) https://doi.org/gc2cbq
DOI: 10.1093/bioinformatics/btx542 · PMID: 28968857 · PMCID: PMC5860199

6. **Toward community standards in the quest for orthologs**
C. Dessimoz, T. Gabaldon, D. S. Roos, E. L. L. Sonnhammer, J. Herrero, A. Altenhoff, R. Apweiler, M. Ashburner, J. Blake, B. Boeckmann, … the Quest for Orthologs Consortium
Bioinformatics (2012-02-12) https://doi.org/f228w5
DOI: 10.1093/bioinformatics/bts050 · PMID: 22332236 · PMCID: PMC3307119

7. **Joining forces in the quest for orthologs**
Toni Gabaldón, Christophe Dessimoz, Julie Huxley-Jones, Albert J Vilella, Erik LL Sonnhammer, Suzanna Lewis
Genome Biology (2009) https://doi.org/frkjfc
DOI: 10.1186/gb-2009-10-9-403 · PMID: 19785718 · PMCID: PMC2768974

8. **SwiftOrtho: A fast, memory-efficient, multiple genome orthology classifier**
Xiao Hu, Iddo Friedberg
GigaScience (2019-10) https://doi.org/ggcr5x
DOI: 10.1093/gigascience/giz118 · PMID: 31648300 · PMCID: PMC6812468

9. **Atlas of protein sequence and structure**
Margaret O Dayhoff
National Biomedical Research Foundation. (1969)

10. **Reduced amino acid alphabets exhibit an improved sensitivity and selectivity in fold assignment**

Eric L. Peterson, Jané Kondev, Julie A. Theriot, Rob Phillips
Bioinformatics (2009-06-01) https://doi.org/btqmnp
DOI: 10.1093/bioinformatics/btp164 · PMID: 19351620 · PMCID: PMC2732308

11. **Fast databank searching with a reduced amino-acid alphabet**
Claudine Landès, Jean-Loup Risler
Bioinformatics (1994) https://doi.org/cvrjmw
DOI: 10.1093/bioinformatics/10.4.453 · PMID: 7804879

12. **RAPSearch: a fast protein similarity search tool for short reads**
Yuzhen Ye, Jeong-Hyeon Choi, Haixu Tang
BMC Bioinformatics (2011-05-15) https://doi.org/dgt5rt
DOI: 10.1186/1471-2105-12-159 · PMID: 21575167 · PMCID: PMC3113943

13. **Simplified amino acid alphabets for protein fold recognition and implications for folding**
Lynne Reed Murphy, Anders Wallqvist, Ronald M. Levy
Protein Engineering, Design and Selection (2000-03) https://doi.org/bdtngh
DOI: 10.1093/protein/13.3.149 · PMID: 10775656

14. **Local homology recognition and distance measures in linear time using compressed amino acid alphabets**
R. C. Edgar
Nucleic Acids Research (2004-01-02) https://doi.org/ckg5d4
DOI: 10.1093/nar/gkh180 · PMID: 14729922 · PMCID: PMC373290

15. **Physical biology of the cell**
Rob Phillips, Julie Theriot, Jane Kondev, Hernan Garcia
Garland Science (2012)

16. **Incomplete nucleic acid sequences**https://www.qmul.ac.uk/sbcs/iubmb/misc/naseq.html

17. **These Are Not the K-mers You Are Looking For: Efficient Online K-mer Counting Using a Probabilistic Data Structure**
Qingpeng Zhang, Jason Pell, Rosangela Canino-Koning, Adina Chuang Howe, C. Titus Brown
PLoS ONE (2014-07-25) https://doi.org/f6kb9b
DOI: 10.1371/journal.pone.0101271 · PMID: 25062443 · PMCID: PMC4111482

18. **The evolution of gene expression levels in mammalian organs**
David Brawand, Magali Soumillon, Anamaria Necsulea, Philippe Julien, Gábor Csárdi, Patrick Harrigan, Manuela Weier, Angélica Liechti, Ayinuer Aximu-Petri, Martin Kircher, … Henrik Kaessmann
Nature (2011-10-19) https://doi.org/fcvk54
DOI: 10.1038/nature10532 · PMID: 22012392

19. **The khmer software package: enabling efficient nucleotide sequence analysis**
Michael R. Crusoe, Hussien F. Alameldin, Sherine Awad, Elmar Boucher, Adam Caldwell, Reed Cartwright, Amanda Charbonneau, Bede Constantinides, Greg Edvenson, Scott Fay, … C. Titus Brown
F1000Research (2015-09-25) https://doi.org/9qp
DOI: 10.12688/f1000research.6924.1 · PMID: 26535114 · PMCID: PMC4608353

20. **UniProt: a worldwide hub of protein knowledge**
rtium

Nucleic Acids Research (2019-01-08) https://doi.org/gfwqck
DOI: 10.1093/nar/gky1049 · PMID: 30395287 · PMCID: PMC6323992

21. **UniProtKB/Swiss-Prot, the Manually Annotated Section of the UniProt KnowledgeBase: How to Use the Entry View**
Emmanuel Boutet, Damien Lieberherr, Michael Tognolli, Michel Schneider, Parit Bansal, Alan J. Bridge, Sylvain Poux, Lydie Bougueleret, Ioannis Xenarios
Methods in Molecular Biology (2016) https://doi.org/f79kbb
DOI: 10.1007/978-1-4939-3167-5_2 · PMID: 26519399

22. **Opisthokont**
Wikipedia
(2020-01-31) https://en.wikipedia.org/w/index.php?title=Opisthokont&oldid=938532808

23. **Sci-Hub provides access to nearly all scholarly literature**
Daniel S Himmelstein, Ariel Rodriguez Romero, Jacob G Levernier, Thomas Anthony Munro, Stephen Reid McLaughlin, Bastian Greshake Tzovaras, Casey S Greene
eLife (2018-03-01) https://doi.org/ckcj
DOI: 10.7554/elife.32822 · PMID: 29424689 · PMCID: PMC5832410

24. **Reproducibility of computational workflows is automated using continuous analysis**
Brett K Beaulieu-Jones, Casey S Greene
Nature biotechnology (2017-04) https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6103790/
DOI: 10.1038/nbt.3780 · PMID: 28288103 · PMCID: PMC6103790

25. **Bitcoin for the biological literature.**
Douglas Heaven
Nature (2019-02) https://www.ncbi.nlm.nih.gov/pubmed/30718888
DOI: 10.1038/d41586-019-00447-9 · PMID: 30718888

26. **Plan S: Accelerating the transition to full and immediate Open Access to scientific publications**
cOAlition S
(2018-09-04) https://www.wikidata.org/wiki/Q56458321

27. **Open access**
Peter Suber
MIT Press (2012)
ISBN: 9780262517638

28. **Open collaborative writing with Manubot**
Daniel S. Himmelstein, Vincent Rubinetti, David R. Slochower, Dongbo Hu, Venkat S. Malladi, Casey S. Greene, Anthony Gitter
Manubot (2020-01-14) https://greenelab.github.io/meta-review/

29. **Opportunities and obstacles for deep learning in biology and medicine**
Travers Ching, Daniel S. Himmelstein, Brett K. Beaulieu-Jones, Alexandr A. Kalinin, Brian T. Do, Gregory P. Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael M. Hoffman, ... Casey S. Greene
Journal of The Royal Society Interface (2018-04-04) https://doi.org/gddkhn
DOI: 10.1098/rsif.2017.0387 · PMID: 29618526 · PMCID: PMC5938574

30. **Open collaborative writing with Manubot**
Daniel S. Himmelstein, Vincent Rubinetti, David R. Slochower, Dongbo Hu, Venkat S. Malladi, Casey S. Greene, Anthony Gitter
PLOS Computational Biology (2019-06-24) https://doi.org/c7np
DOI: 10.1371/journal.pcbi.1007128 · PMID: 31233491 · PMCID: PMC6611653