Manuscript Title

This manuscript (permalink) was automatically generated from taylorreiter/2022-paper-charcoal@91342b1 on June 15, 2022.

Authors

- **John Doe**
  XXXX-XXXX-XXXX-XXXX · johndoe · johndoe
  Department of Something, University of Whatever · Funded by Grant XXXXXXXX

- **Jane Roe**
  XXXX-XXXX-XXXX-XXXX · janeroe
  Department of Something, University of Whatever; Department of Whatever, University of Something

Abstract

Metagenomics has expanded our knowledge of microbial diversity, but contaminant sequences are frequently accidentally included in metagenome-assembled genomes. Genome contamination is often estimated by the presence of marker genes that are biased against detecting contaminants lacking these sequences. Further, most contamination detection tools do not remove contamination. We present charcoal, a tool that rapidly identifies and removes contamination in metagenome-assembled genomes using k-mer based methods. K-mers are nucleotide sequences of length k. Sufficiently long k-mers are usually specific to a taxonomic lineage. Taking advantage of this property of k-mers, charcoal identifies majority and minority lineages for each contiguous sequence in a genome and removes contiguous sequences belonging to minority lineages when those lineages occur below a taxonomic threshold (by default, order). Applying charcoal to the Genome Taxonomy Database rs207, we found approximately XX% of genomes in GTDB were contaminated, with contamination broadly distributed across species and occurring in representative and RefSeq genomes. Genomes with longer contiguous sequences were less likely to be contaminated. Our results show concordance with CheckM on detecting the presence of contamination in a genome. Charcoal is a snakemake workflow developed around the tool sourmash. It is available at github.com/dib-lab/charcoal, and is pip installable.

This manuscript is a template (aka "rootstock") for Manubot, a tool for writing scholarly manuscripts. Use this template as a starting point for your manuscript.

The rest of this document is a full list of formatting elements/features supported by Manubot. Compare the input (`.md` files in the `/content` directory) to the output you see below.

Basic formatting

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
two spaces at end of line.

Document organization

Document section headings:

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

A heading centered on its own printed page

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 [1].

Citation by PubMed Central ID [2].

Citation by PubMed ID [3].

Citation by Wikidata ID [4].

Citation by ISBN [5].

Citation by URL [6].

Citation by alias [7].

Multiple citations can be put inside the same set of brackets [1,5,7]. Manubot plugins provide easier, more convenient visualization of and navigation between citations [2,3,7,8].

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

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

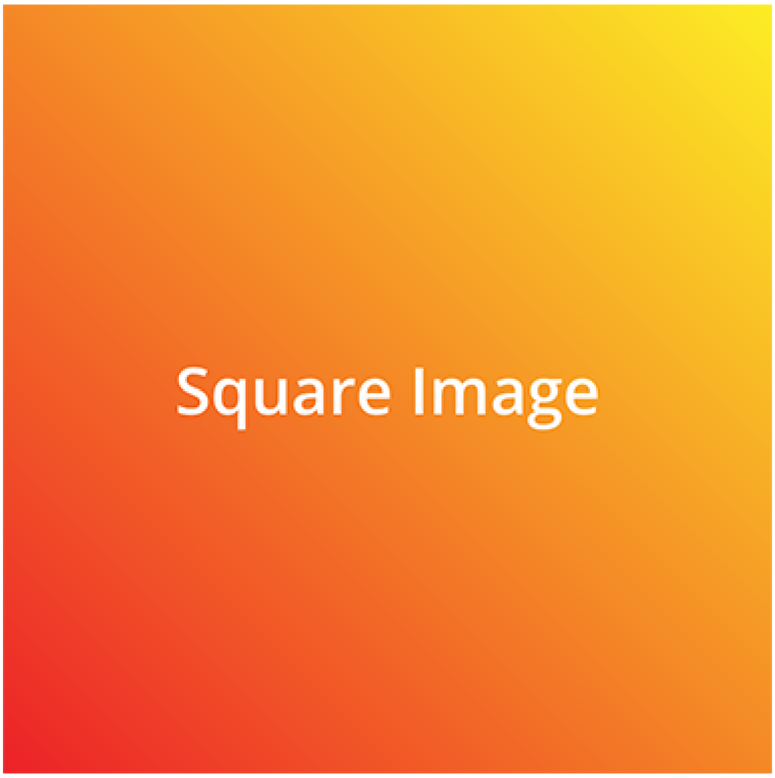

Figure 1: A square image at actual size and with a bottom caption. Loaded from the latest version of image on GitHub.

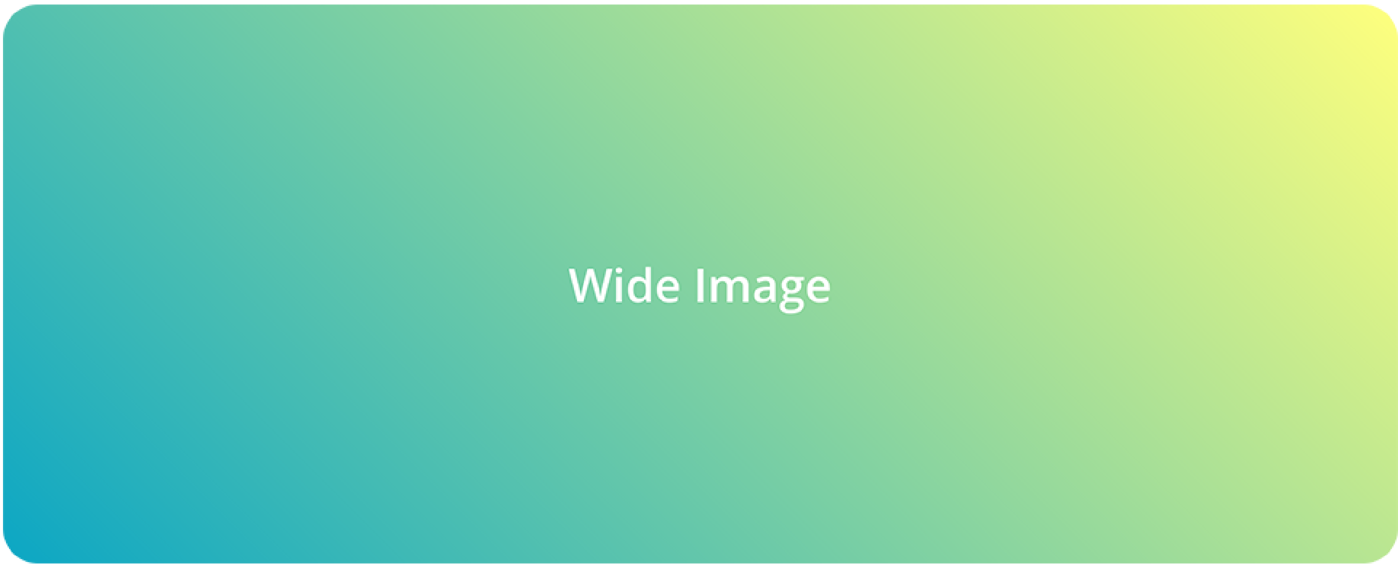

Figure 2: An image too wide to fit within page at full size. Loaded from a specific (hashed) version of the image on GitHub.

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	2497757247093699959574966967627724	0766303535475945713821785251664274	`nasa.gov`

Table 3: A table with merged 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:

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

Introduction

Metagenomic sequencing has expanded our knowledge of microbial communities and their diversity [9,10,11]. *De novo* metagenome analysis has generated thousands of draft genomes, termed metagenome-assembled genomes (MAGs), from organisms from diverse environments [10,12,13,14]. Recently, large-scale re-analysis efforts have led to a rapid expansion in draft genomes in public repositories like the European Nucleotide Archive [14] and the Joint Genome Institute IMG/M [10]. Increased observation of draft genomes across the tree of life better enables researchers to contextualize new sequencing data and the roles that microorganisms play in diverse metabolic processes [15,16].

MAG inference relies on assembly and binning of metagenomic sequencing data. Assembly produces long contiguous sequences by identifying overlaps between short sequencing reads, while binning groups assembled sequences into MAGs using read coverage and tetramernucleotide frequency. Both processes are subject to biases that can reduce the completeness of or increase the contamination in a MAG: low sequencing coverage or high genomic variation causes short read assemblers to break contiguous sequences into shorter pieces (CITE), which decreases the signal for and accuracy of binning (CITE). Commonly, the completeness and purity of MAGs is estimated through the presence and sequence composition of single-copy marker genes [17] [18], with MAGs that reach >90% completeness and <5% contamination considered high quality [17]. Single-copy marker genes are sets of genes that are present once in a genome of almost all members of a taxonomic group [17,19]. Using these genes to estimate contamination leads to two important biases. First, given the assumption that marker genes are universally present in genomes, if a marker gene resides on a contaminant sequence but no other sequence for that marker gene is present in the genome, it will not be detected as contamination [17,20]. When a MAG is substantially complete, this may lead to a small underestimation in contamination (~3% [17]), but as completeness decreases, contamination may be substantially underestimated [20]. Second, contiguous sequences which do not contain marker sequences are not included among contamination estimates [17] (note checkm called out plasmids and phages for this bias specifically).

Given these biases, methods that do not rely solely on marker genes may be better suited for contamination estimation. Many tools have recently been developed that rely on different strategies for detecting contamination [21]. GUNC expands beyond marker genes and uses the full complement of genes in a genome to identify contiguous sequences that fall outside of the predicted lineage [22]. It estimates the distribution of taxonomic assignments within and across contiguous sequences to identify contaminants even among sequences that are poorly taxonomically labelled [22]. Conterminator detects cross-kingdom contamination using all-vs-all alignment and is geared toward contamination detection in large collections of genomes (e.g. databases) [23]. Sequences are considered contaminants when at least 100 nucleotides and no more than 20 kb share a sequence identity of at least 90% in genomes that originate from different kingdoms [23]. In addition, conterminator finds contamination in protein sequences by identifying protein clusters that contain sequences from cross-kingdom members [23].

Removing contamination is a separate problem from estimating its extent. Tools like RefineM [15], MAGPpurify [24], and BlobTools [25] rely on a combination of GC content, tetramernucleotide frequency, read coverage, phylogenetic or clade markers, conspecific sequences, and known contaminants to flag contaminant contiguous sequences for removal. The inclusion of read coverage profiles necessitates the use of the original sequencing reads to produce coverage profiles in BAM format for the genomes. Given that sequencing data and BAM files are usually orders of magnitude larger than genome sequences, this requirement limits the utility of such approaches, especially for large-scale genome databases. Tools like GUNC are poised for contamination removal without relying on read coverage profiles, but as of yet such approaches are unimplemented [22].

Long k-mers capture relatedness between organisms, where a k=31 captures species-level similarity [26]. K-mers offer an alternative metric to identify contamination, especially in sequences lacking marker genes. Here we describe Charcoal, an automated method for filtering contaminant contiguous sequences from MAGs. We show... We show... We envisage that charcoal will complement marker gene-based approaches for contamination estimation, removing problematic sequences before they are further analyzed or propagated in public databases.

Methods

Overview

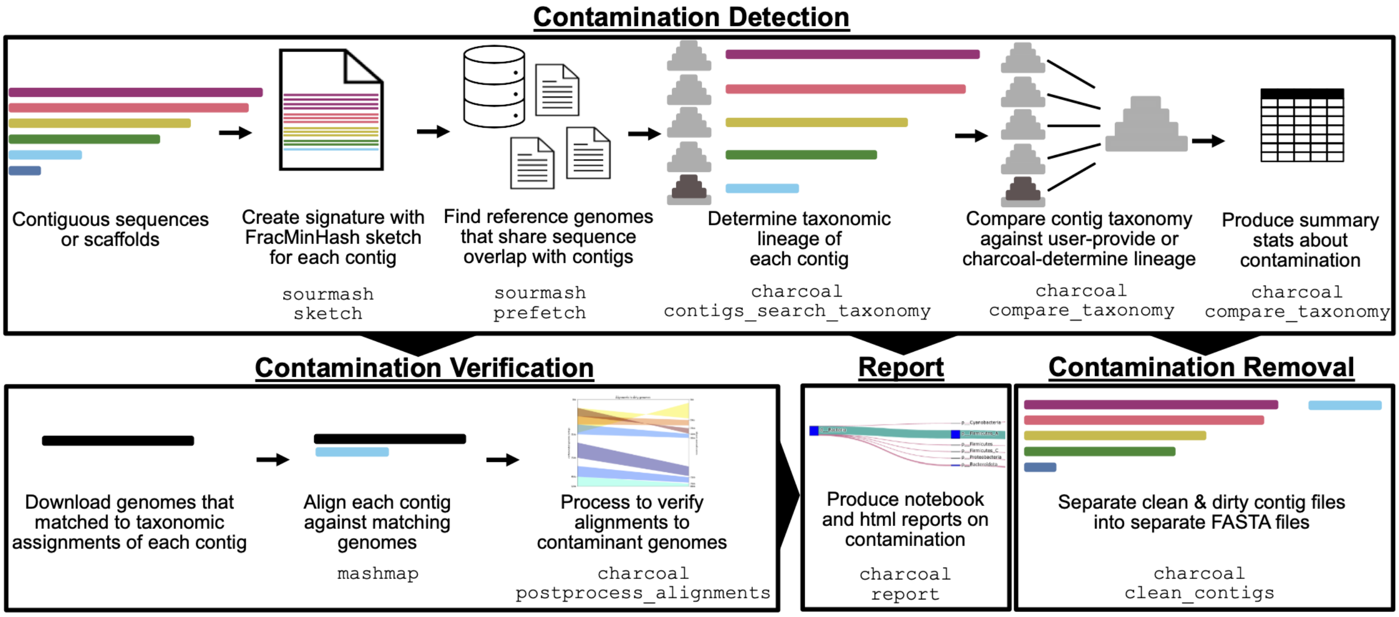
Contamination Detection
Contiguous sequences or scaffolds
Create signature with FracMinHash sketch for each contig
sourmash sketch
Find reference genomes that share sequence overlap with contigs
sourmash prefetch
Determine taxonomic lineage of each contig
charcoal contigs_search_taxonomy
Compare contig taxonomy against user-provide or charcoal-determine lineage
charcoal compare_taxonomy
Produce summary stats about contamination
charcoal compare_taxonomy
Contamination Verification
Download genomes that matched to taxonomic assignments of each contig
Align each contig against matching genomes
mashmap
Process to verify alignments to contaminant genomes
charcoal postprocess_alignments
Report
Produce notebook and html reports on contamination
charcoal report
Contamination Removal
Separate clean & dirty contig files into separate FASTA files
charcoal clean_contigs

Figure 5: Summary of steps used to decontaminate genomes with charcoal. In the first stage of the pipeline, contamination detection occurs by comparing the taxonomic assignment of each contig in a genome against all other contigs. Contigs with inconsistent taxonomy are flagged as contaminants, by default if they differ at the order level or above. The results of this step are summarized to report the contaminating genomes, the number of contaminant contigs and basepairs at each level of taxonomy, and other metrics like the fraction of the genome that was identifiable and the fraction that was assignable to the majority lineage. The outputs of the first stage of the pipeline can then be used in any of three additional reporting steps. First, contamination can be verified by aligning the contaminant contigs against their identified matching reference genome, and the mappings can be visualized. Second, an html document can be produced that gives an overview of the contamination in each genome. Lastly, the contaminant contigs can be separated out from the non-contaminant contigs, writing two separate FASTA files per genome.

To identify contamination, charcoal first creates a FracMinHash sketch for each contiguous sequence ("contig") in an input genome. Charcoal then identifies all genomes in a database ("reference genomes") that share sequence overlap with the input genome using sourmash `prefetch`. Subsequent operations subset the original database to include only the genomes identified by `prefetch`, reducing search volumes. Using these matches, charcoal uses sourmash `gather` to identify the minimum set of genomes that cover (or contain) the k-mers in each contig [27]. Charcoal then determines the taxonomic lineage of each contig using a lineage spreadsheet that records the taxonomy of each reference genome in the database; the taxonomic assignment occurs at the lowest common ancestor of all taxonomic assignments given to a contig. If there is an exact match between the input genome and a genome in the database, this match is removed to allow decontamination to continue.

Charcoal compares the taxonomic lineage of each contig against the lineage of the input genome. If the contig has a different lineage before or at the configured taxonomic rank (order by default) than that of the majority lineage, the contig is considered a contaminant. For each contig, charcoal reports the fraction of identified hashes (total and to each major and minor lineage), an estimate of contaminated base pairs (at each taxonomic rank match), as well as the fraction of contigs and base pairs unable to be identified.

The input genome lineage can be user-provided, or it can be determined by charcoal via majority vote of all lineages assigned to all contigs. If the lineage is determined by charcoal, by default a minimum 10% of the input genome must have been assigned a taxonomic lineage, and 20% of assigned sequences must match to the majority lineage. If these specifications are not met, charcoal will not decontaminate the input genome unless the user specifies a lineage. The user can optionally specify a lineage (e.g. d__Eukaryota), and charcoal will remove contigs that have a lineage different from the user-specified one. This allows charcoal to remove contigs from an input genome when some contigs from that genome occur in the database and when the input genome is not related to anything in a database. Charcoal reports whether the provided lineage agrees with k-mer classification at or above the genus level.

After the initial stage of contaminant detection, charcoal can perform additional tasks to verify, summarize, or remove the contaminant sequences. Contaminant verification downloads the reference genome sequence for any genome that was detected among the input genome contigs and aligns the contigs against those genomes using mashmap [28]. Contaminant removal separates "clean" from "dirty" contigs and outputs each set into a FASTA file. Importantly, charcoal will not remove a contig if it is unidentifiable, whether it is too short to be sketched or does not contain sequences in the reference database. While these contigs could still be contaminants, charcoal assumes contigs are clean for which it has no information. Therefore, charcoal will fail to detect contamination for very short contigs which contain no selected k-mers, as well as contigs with novel DNA content. Lastly, charcoal has a report feature that summarizes and visualizes the taxonomic lineages detected in each input genome as well as the alignments between the input genome and reference genomes.

Sourmash `prefetch` is the most compute intensive step in the decontamination process. When using the GTDB rs207 representative database, this step does not exceed 8GB of RAM.

- statement that charcoal is database dependent (probably can go in last paragraph with RAM/CPU considerations)

Availability and dependencies

Charcoal is written in python3 and can be installed via pip as charcoal-bio. The core algorithms (contaminant detection and removal) depend on sourmash and snakemake. Contaminant verification depends on lxml and mashmap [28]. Reporting depends on mummer, papermill, notebook, and plotly [29]. The source code is available at github.com/dib-lab/charcoal. ZENODO DOI.

Datasets and benchmarking

Results

old results outline: + charcoal estimates low contamination in non-representative GTDB genomes + charcoal assigns correct taxonomy to all non-representative GTDB genomes + charcoal vs. checkm + charcoal vs. checkm: mgnify, tara + checkm on charcoal clean + prokka on charcoal dirty + charcoal vs. refineM and magpurify + verification of contamination/contam case studies + case studies + user-provided lineages + cDBG/spacegraphcats? + user-specified lineages

Discussion

Benefits and drawbacks of charcoal.

Commonly, the completeness and contamination of MAGs is estimated through the presence and sequence composition of single-copy marker genes [17,18], with MAGs that reach >90% completeness and <5% contamination considered high quality [17]. Using these genes to estimate contamination leads to two important biases. First, given the assumption that marker genes are universally present in genomes, if a marker gene resides on a contaminant sequence but no other sequence for that marker gene is present in the genome, it will not be detected as contamination [17,20]. When a MAG is substantially complete, this may lead to a small underestimation in contamination (~3% [17]), but as completeness decreases, contamination may be substantially underestimated [20]. Second, contiguous sequences which do not contain marker sequences are not included among contamination estimates [17]. Charcoal avoids these pitfalls by using k-mers, which are sampled evenly across the genome [27]. We envisage charcoal acting as a complementary analysis tool to CheckM.

While charcoal avoids the above biases, it is itself biased against detecting contamination in very short contigs. By default, a minimum of three k-mers must be taxonomically annotated on a contig for a contig to be removed; because contigs are first sketched, many short contigs will not contain three k-mers.

Future work

- strain heterogeneity estimation or strain insertion in newick tree using prefetch & clean results
- completeness estimation, potentially via comparison to conspecifics via genome size
- database with eukaryotic sequences
- estimation via protein sequences for divergent organisms not well represented in databases

Notes

- fastani is an accepted way to do average nucleotide identity calculate, and it relies on k-mers.
- from checkm paper: > Bias in genome quality estimates: Quality estimates based on individual marker genes or collocated marker sets exhibit a bias resulting in completeness being

overestimated and contamination being underestimated. This bias is the result of marker genes residing on foreign DNA that are otherwise absent in a genome being mistakenly interpreted as an indication of increased completeness as opposed to contamination.

- we don’t deal with strain heterogeneity, as this occurs below the species-level aggregation in the LCA

References

1. **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

2. **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

3. **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

4. **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

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

6. **Open collaborative writing with Manubot**
Daniel S Himmelstein, Vincent Rubinetti, David R Slochower, Dongbo Hu, Venkat S Malladi, Casey S Greene, Anthony Gitter
Manubot (2020-05-25) https://greenelab.github.io/meta-review/

7. **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) https://doi.org/gddkhn
DOI: 10.1098/rsif.2017.0387 · PMID: 29618526 · PMCID: PMC5938574

8. **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

9. **A new view of the tree of life**
Laura A Hug, Brett J Baker, Karthik Anantharaman, Christopher T Brown, Alexander J Probst, Cindy J Castelle, Cristina N Butterfield, Alex W Hernsdorf, Yuki Amano, Kotaro Ise, … Jillian F Banfield
Nature Microbiology (2016-04-11) https://doi.org/bpkh
DOI: 10.1038/nmicrobiol.2016.48 · PMID: 27572647

10. **A genomic catalog of Earth’s microbiomes**

Stephen Nayfach, Simon Roux, Rekha Seshadri, Daniel Udwary, Neha Varghese, Frederik Schulz, Dongying Wu, David Paez-Espino, I-Min Chen, Marcel Huntemann, …
Nature Biotechnology (2020-11-09) https://doi.org/ghjh4b
DOI: 10.1038/s41587-020-0718-6 · PMID: 33169036 · PMCID: PMC8041624

11. **Metagenomic compendium of 189,680 DNA viruses from the human gut microbiome**
Stephen Nayfach, David Páez-Espino, Lee Call, Soo Jen Low, Hila Sberro, Natalia N Ivanova, Amy D Proal, Michael A Fischbach, Ami S Bhatt, Philip Hugenholtz, Nikos C Kyrpides
Nature Microbiology (2021-06-24) https://doi.org/gkx3md
DOI: 10.1038/s41564-021-00928-6 · PMID: 34168315 · PMCID: PMC8241571

12. **Community structure and metabolism through reconstruction of microbial genomes from the environment**
Gene W Tyson, Jarrod Chapman, Philip Hugenholtz, Eric E Allen, Rachna J Ram, Paul M Richardson, Victor V Solovyev, Edward M Rubin, Daniel S Rokhsar, Jillian F Banfield
Nature (2004-02-01) https://doi.org/b85j5j
DOI: 10.1038/nature02340 · PMID: 14961025

13. **Extensive Unexplored Human Microbiome Diversity Revealed by Over 150,000 Genomes from Metagenomes Spanning Age, Geography, and Lifestyle**
Edoardo Pasolli, Francesco Asnicar, Serena Manara, Moreno Zolfo, Nicolai Karcher, Federica Armanini, Francesco Beghini, Paolo Manghi, Adrian Tett, Paolo Ghensi, … Nicola Segata
Cell (2019-01) https://doi.org/gfthv3
DOI: 10.1016/j.cell.2019.01.001 · PMID: 30661755 · PMCID: PMC6349461

14. **A unified catalog of 204,938 reference genomes from the human gut microbiome**
Alexandre Almeida, Stephen Nayfach, Miguel Boland, Francesco Strozzi, Martin Beracochea, Zhou Jason Shi, Katherine S Pollard, Ekaterina Sakharova, Donovan H Parks, Philip Hugenholtz, … Robert D Finn
Nature Biotechnology (2020-07-20) https://doi.org/gg5hgn
DOI: 10.1038/s41587-020-0603-3 · PMID: 32690973 · PMCID: PMC7801254

15. **Recovery of nearly 8,000 metagenome-assembled genomes substantially expands the tree of life**
Donovan H Parks, Christian Rinke, Maria Chuvochina, Pierre-Alain Chaumeil, Ben J Woodcroft, Paul N Evans, Philip Hugenholtz, Gene W Tyson
Nature Microbiology (2017-09-11) https://doi.org/cczd
DOI: 10.1038/s41564-017-0012-7 · PMID: 28894102

16. **proGenomes2: an improved database for accurate and consistent habitat, taxonomic and functional annotations of prokaryotic genomes**
Daniel R Mende, Ivica Letunic, Oleksandr M Maistrenko, Thomas SB Schmidt, Alessio Milanese, Lucas Paoli, Ana Hernández-Plaza, Askarbek N Orakov, Sofia K Forslund, Shinichi Sunagawa, … Peer Bork
Nucleic Acids Research (2019-10-24) https://doi.org/ggctnd
DOI: 10.1093/nar/gkz1002 · PMID: 31647096 · PMCID: PMC7145564

17. **CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes**
Donovan H Parks, Michael Imelfort, Connor T Skennerton, Philip Hugenholtz, Gene W Tyson
Genome Research (2015-05-14) https://doi.org/bb8q
DOI: 10.1101/gr.186072.114 · PMID: 25977477 · PMCID: PMC4484387

18. **BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs**

Felipe A Simão, Robert M Waterhouse, Panagiotis Ioannidis, Evgenia V Kriventseva, Evgeny M Zdobnov
Bioinformatics (2015-06-09) https://doi.org/gfznpw
DOI: 10.1093/bioinformatics/btv351 · PMID: 26059717

19. **PhyloSift: phylogenetic analysis of genomes and metagenomes**
Aaron E Darling, Guillaume Jospin, Eric Lowe, Frederick A Matsen IV, Holly M Bik, Jonathan A Eisen
PeerJ (2014-01-09) https://doi.org/gddvvt
DOI: 10.7717/peerj.243 · PMID: 24482762 · PMCID: PMC3897386

20. **Rokubacteria: Genomic Giants among the Uncultured Bacterial Phyla**
Eric D Becraft, Tanja Woyke, Jessica Jarett, Natalia Ivanova, Filipa Godoy-Vitorino, Nicole Poulton, Julia M Brown, Joseph Brown, MCY Lau, Tullis Onstott, … Ramunas Stepanauskas
Frontiers in Microbiology (2017-11-28) https://doi.org/gcn68w
DOI: 10.3389/fmicb.2017.02264 · PMID: 29234309 · PMCID: PMC5712423

21. **Contamination detection in genomic data: more is not enough**
Luc Cornet, Denis Baurain
Genome Biology (2022-02-21) https://doi.org/gpkc9r
DOI: 10.1186/s13059-022-02619-9 · PMID: 35189924 · PMCID: PMC8862208

22. **GUNC: detection of chimerism and contamination in prokaryotic genomes**
Askarbek Orakov, Anthony Fullam, Luis Pedro Coelho, Supriya Khedkar, Damian Szklarczyk, Daniel R Mende, Thomas SB Schmidt, Peer Bork
Genome Biology (2021-06-13) https://doi.org/gk5s96
DOI: 10.1186/s13059-021-02393-0 · PMID: 34120611 · PMCID: PMC8201837

23. **Terminating contamination: large-scale search identifies more than 2,000,000 contaminated entries in GenBank**
Martin Steinegger, Steven L Salzberg
Genome Biology (2020-05-12) https://doi.org/ggww3q
DOI: 10.1186/s13059-020-02023-1 · PMID: 32398145 · PMCID: PMC7218494

24. **New insights from uncultivated genomes of the global human gut microbiome**
Stephen Nayfach, Zhou Jason Shi, Rekha Seshadri, Katherine S Pollard, Nikos C Kyrpides
Nature (2019-03-13) https://doi.org/gfwwg2
DOI: 10.1038/s41586-019-1058-x · PMID: 30867587 · PMCID: PMC6784871

25. **BlobTools: Interrogation of genome assemblies**
Dominik R Laetsch, Mark L Blaxter
F1000Research (2017-07-31) https://doi.org/gffk7z
DOI: 10.12688/f1000research.12232.1

26. **MetaPalette: a <i>k</i> -mer Painting Approach for Metagenomic Taxonomic Profiling and Quantification of Novel Strain Variation**
David Koslicki, Daniel Falush
mSystems (2016-06-28) https://doi.org/gg3gbd
DOI: 10.1128/msystems.00020-16 · PMID: 27822531 · PMCID: PMC5069763

27. **Lightweight compositional analysis of metagenomes with FracMinHash and minimum metagenome covers**
Luiz Irber, Phillip T Brooks, Taylor Reiter, NTessa Pierce-Ward, Mahmudur Rahman Hera, David Koslicki, CTitus Brown
Cold Spring Harbor Laboratory (2022-01-12) https://doi.org/gn34zt

DOI: 10.1101/2022.01.11.475838

28. **A fast adaptive algorithm for computing whole-genome homology maps**
Chirag Jain, Sergey Koren, Alexander Dilthey, Adam M Phillippy, Srinivas Aluru
Bioinformatics (2018-09-01) https://doi.org/gd9ndx
DOI: 10.1093/bioinformatics/bty597 · PMID: 30423094 · PMCID: PMC6129286

29. **Using <scp>MUMmer</scp> to Identify Similar Regions in Large Sequence Sets**
Arthur L Delcher, Steven L Salzberg, Adam M Phillippy
Current Protocols in Bioinformatics (2003-01) https://doi.org/dks6c5
DOI: 10.1002/0471250953.bi1003s00 · PMID: 18428693