Genome assembly, a universal theoretical framework: unifying and generalizing the safe and complete algorithms

Massimo Cairo, Shahbaz Khan, Romeo Rizzi, Sebastian Schmidt,

Alexandru I. Tomescu and Elia C. Zirondelli

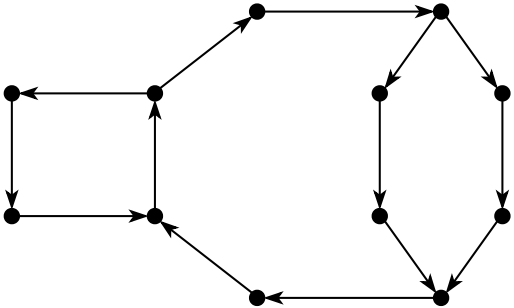
Outline

Genome Assembly & Safety

Models of genome assembly

Genome graph:

- Used by almost all modern assemblers
- Each edge is part of the genome
- The genome is a walk



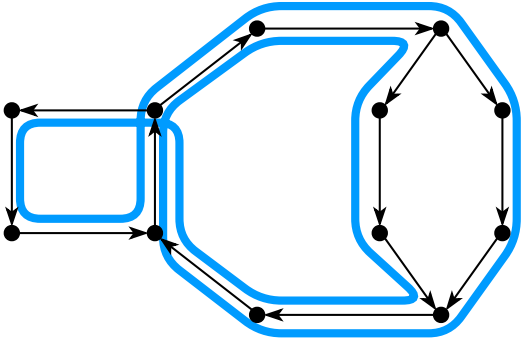
Models of genome assembly

Genome graph:

- Used by almost all modern assemblers
- Each edge is part of the genome
- The genome is a walk

Additional information:

- There is a single circular genome



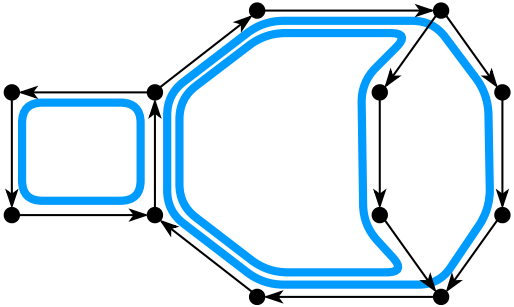
Models of genome assembly

Genome graph:

- Used by almost all modern assemblers
- Each edge is part of the genome
- The genome is a walk

Additional information:

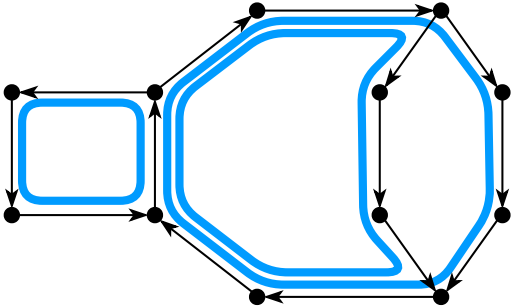
- There are multiple circular genomes



Finding the true genome

Challenge:

- Infinitely many possible solutions
  What is the correct one?



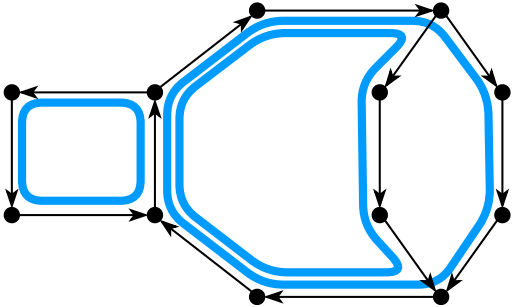
Finding the true genome

Challenge:

- Infinitely many possible solutions
  What is the correct one?

Solution:

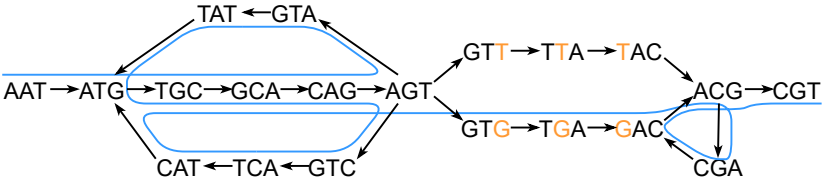
- Find only safe subwalks
  of the true genome
- But be as complete as possible



Accounting for practical issues

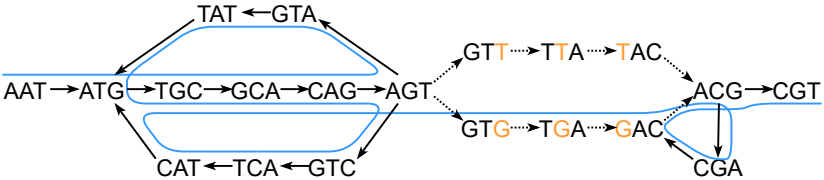
Father: AATGCAGTATGCAGTCATGCAGTTACGACGT
Mother: AATGCAGTATGCAGTCATGCAGTGACGACGT

Accounting for practical issues



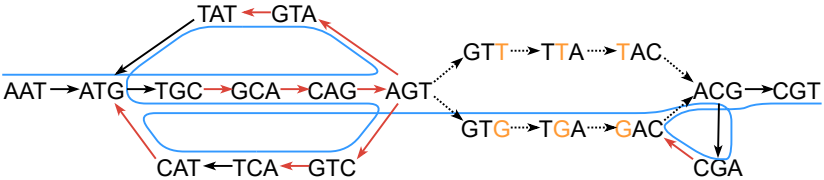
Father: AATGCAGTATGCAGTCATGCAGTTACGACGT
Mother: AATGCAGTATGCAGTCATGCAGTGACGACGT

Accounting for practical issues



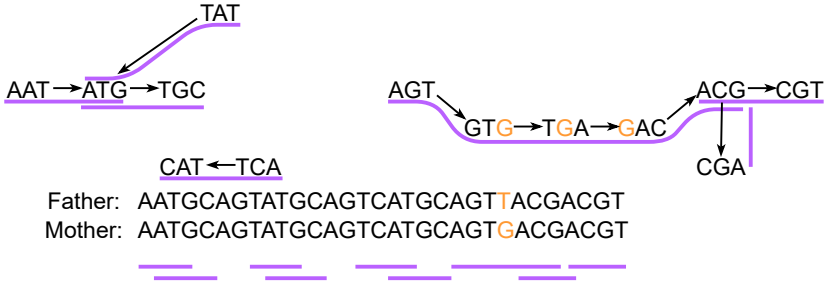
Father: AATGCAGTATGCAGTCATGCAGTTACGACGT
Mother: AATGCAGTATGCAGTCATGCAGTGACGACGT

Accounting for practical issues

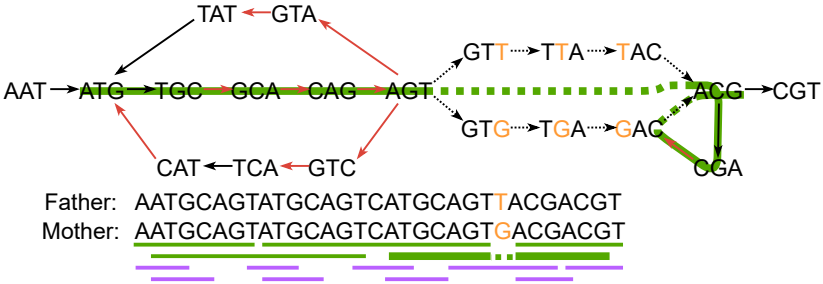


Father: AATGCAGTATGCAGTCATGCAGTTACGACGT
Mother: AATGCAGTATGCAGTCATGCAGTGACGACGT

Accounting for practical issues



Accounting for practical issues

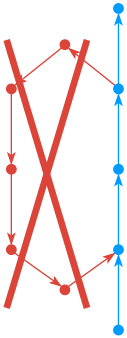


Current Safe and Complete Algorithms

Properties of current safe and complete algorithms

Single circular, Omnitigs:
characterisation (Tomescu and Medvedev, 2017)
optimal $O(mn)$ (Cairo et al., 2019)
output optimal $O(m + n + o)$ (Cairo et al., 2020b)

Multi circular:
$O(m^2 n)$ (Acosta, Mäkinen, and Tomescu, 2018)



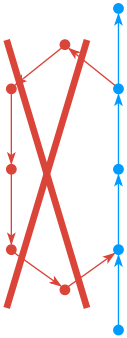
Properties of current safe and complete algorithms

Single circular: $O(mn)$, $O(m+n+o)$
Multi circular: $O(m^2 n)$

But many other models are relevant:

- k solution walks
- linear models
- partial coverage
- partial visibility

Unified theory for all combinations of these?



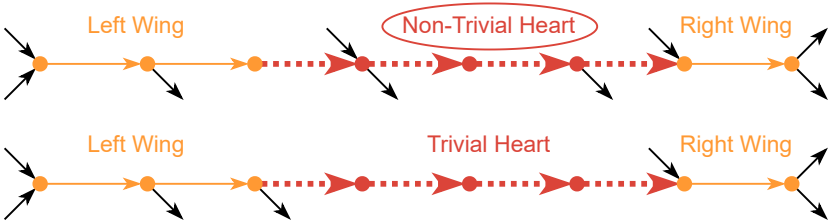
Unifying the Theory: the Hydrostructure (Cairo et al., 2020a)

Unification: the hydrostructure of a walk W

- Defined on a strongly connected graph $G := V \cup E$ and a walk $W \subseteq G$

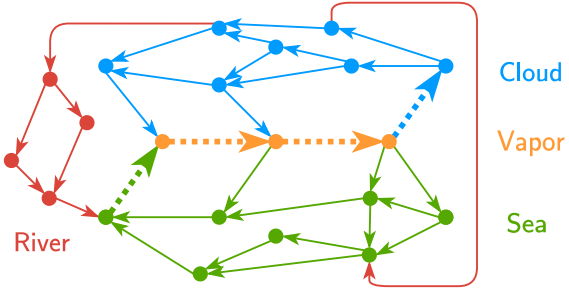
Unification: the hydrostructure of a walk W

- Defined on a strongly connected graph $G := V \cup E$ and a walk $W \subseteq G$
- In this talk: only non-trivial hearts of paths (without repetitions of nodes/edges)



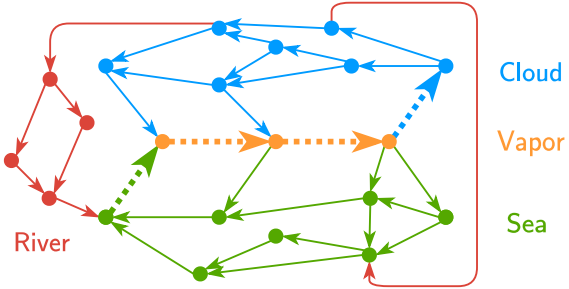
Unification: the hydrostructure of a walk W

- Decomposes the graph into four regions



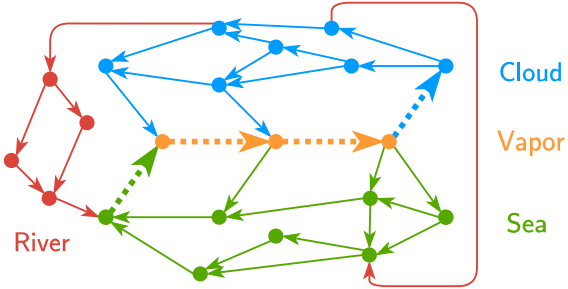
Unification: the hydrostructure of a walk W

- Decomposes the graph into four regions
- The connectivity between regions is restricted



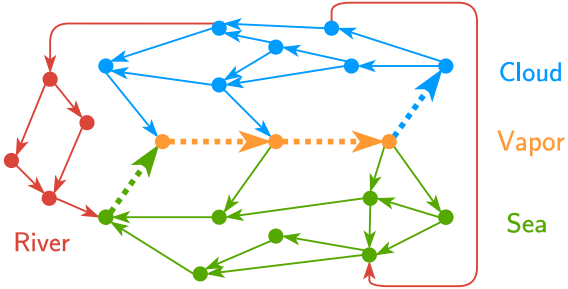
Unification: the hydrostructure of a walk W

- Decomposes the graph into four regions
- The connectivity between regions is restricted
- If Vapor is a path, then W is a Sea-Cloud bottleneck



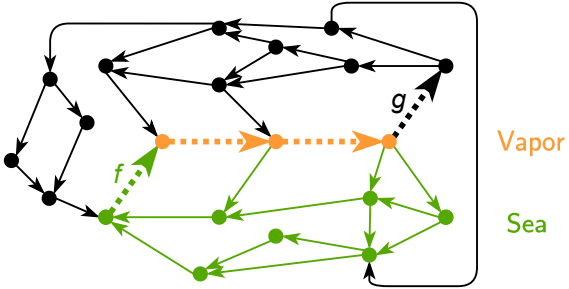
Unification: the hydrostructure of a walk W

- Decomposes the graph into four regions
- The connectivity between regions is restricted
- If Vapor is a path, then W is a Sea-Cloud bottleneck
- For any model: if each solution walk goes from Sea to Cloud, then W is safe



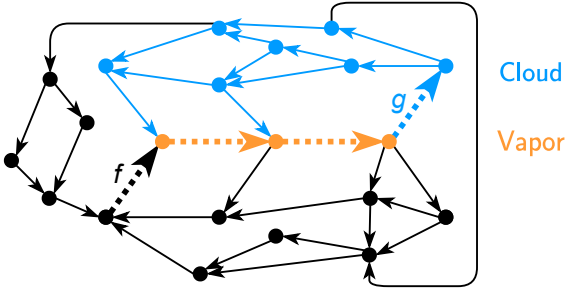
Definition of the hydrostructure

- W is a walk from edge f to edge g
- $R^+(W)$ is everything reachable from f in $G \setminus g$



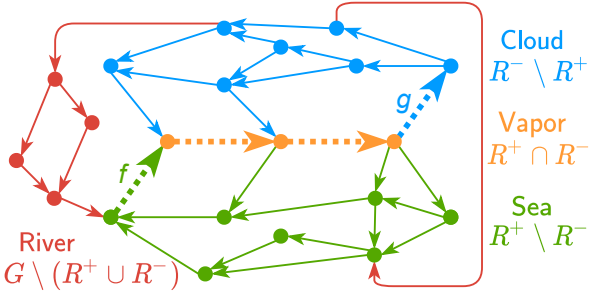
Definition of the hydrostructure

- W is a walk from edge f to edge g
- $R^+(W)$ is everything reachable from f in $G \setminus g$
- $R^-(W)$ is everything backwards-reachable from g in $G \setminus f$



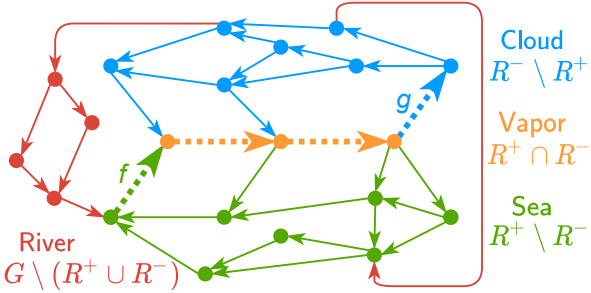
Definition of the hydrostructure

- W is a walk from edge f to edge g
- $R^+(W)$ is everything reachable from f in $G \setminus g$
- $R^-(W)$ is everything backwards-reachable from g in $G \setminus f$



Definition of the hydrostructure

- W is a walk from edge f to edge g
- $R^+(W)$ is everything reachable from f in $G \setminus g$
- $R^-(W)$ is everything backwards-reachable from g in $G \setminus f$
- W is a Sea-Cloud bottleneck if and only if **Vapor is a path**

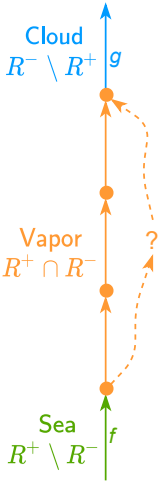


The bottleneck property

Definition 1: $R^+(W) := \{x \in G \mid f \to x \in G \setminus g\}$
Definition 2: $R^-(W) := \{x \in G \mid x \to g \in G \setminus f\}$
Definition 3: W is a Sea-Cloud bottleneck if each walk from Sea to Cloud has W as subwalk

Lemma: W is a Sea-Cloud bottleneck $\iff$ Vapor is a path

- f is the only way to enter $R^-(W)$ (by Def. 2)

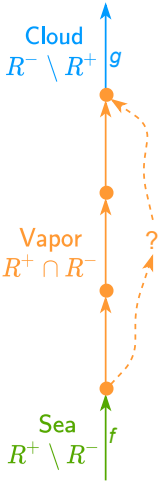


The bottleneck property

Definition 1: $R^+(W) := \{x \in G \mid f \to x \in G \setminus g\}$
Definition 2: $R^-(W) := \{x \in G \mid x \to g \in G \setminus f\}$
Definition 3: W is a Sea-Cloud bottleneck if each walk from Sea to Cloud has W as subwalk

Lemma: W is a Sea-Cloud bottleneck $\iff$ Vapor is a path

- f is the only way to enter $R^-(W)$ (by Def. 2)
- g is the only way to exit $R^+(W)$ (by Def. 1)

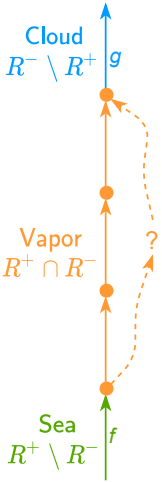


The bottleneck property

Definition 1: $R^+(W) := \{x \in G \mid f \to x \in G \setminus g\}$
Definition 2: $R^-(W) := \{x \in G \mid x \to g \in G \setminus f\}$
Definition 3: W is a Sea-Cloud bottleneck if each walk from Sea to Cloud has W as subwalk

Lemma: W is a Sea-Cloud bottleneck $\iff$ Vapor is a path

- f is the only way to enter $R^-(W)$ (by Def. 2)
- g is the only way to exit $R^+(W)$ (by Def. 1)
- $\Rightarrow$ Sea-Cloud walks have a head(f)-tail(g) subwalk X

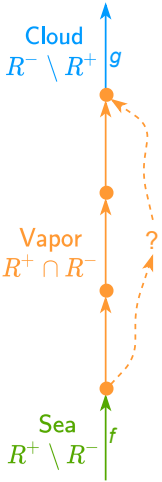


The bottleneck property

Definition 1: $R^+(W) := \{x \in G \mid f \to x \in G \setminus g\}$
Definition 2: $R^-(W) := \{x \in G \mid x \to g \in G \setminus f\}$
Definition 3: W is a Sea-Cloud bottleneck if each walk from Sea to Cloud has W as subwalk

Lemma: W is a Sea-Cloud bottleneck $\iff$ Vapor is a path

- f is the only way to enter $R^-(W)$ (by Def. 2)
- g is the only way to exit $R^+(W)$ (by Def. 1)
- $\Rightarrow$ Sea-Cloud walks have a head(f)-tail(g) subwalk X
- A minimal one is in Vapor



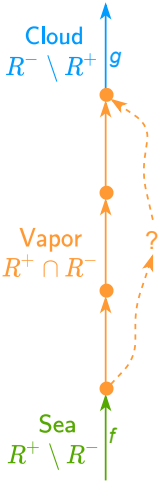
The bottleneck property

Definition 1: $R^+(W) := \{x \in G \mid f \to x \in G \setminus g\}$
Definition 2: $R^-(W) := \{x \in G \mid x \to g \in G \setminus f\}$
Definition 3: W is a Sea-Cloud bottleneck if each walk from Sea to Cloud has W as subwalk

Lemma: W is a Sea-Cloud bottleneck $\iff$ Vapor is a path

- f is the only way to enter $R^-(W)$ (by Def. 2)
- g is the only way to exit $R^+(W)$ (by Def. 1)
- $\Rightarrow$ Sea-Cloud walks have a head(f)-tail(g) subwalk X
- A minimal one is in Vapor

$\Rightarrow$ If W Sea-Cloud bottleneck, then $fXg = W$, so Vapor is a path



The bottleneck property

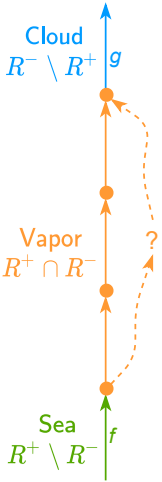
Definition 1: $R^+(W) := \{x \in G \mid f \to x \in G \setminus g\}$
Definition 2: $R^-(W) := \{x \in G \mid x \to g \in G \setminus f\}$
Definition 3: W is a Sea-Cloud bottleneck if each walk from Sea to Cloud has W as subwalk

Lemma: W is a Sea-Cloud bottleneck $\iff$ Vapor is a path

- f is the only way to enter $R^-(W)$ (by Def. 2)
- g is the only way to exit $R^+(W)$ (by Def. 1)
- $\Rightarrow$ Sea-Cloud walks have a head(f)-tail(g) subwalk X
- A minimal one is in Vapor

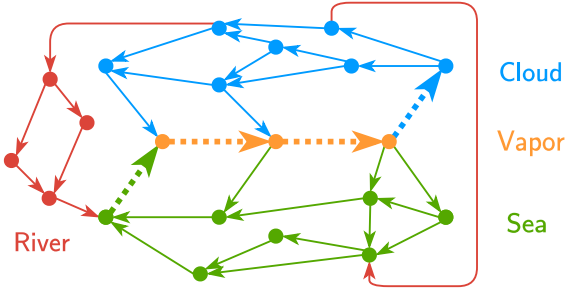
$\Rightarrow$ If W Sea-Cloud bottleneck, then $fXg = W$, so Vapor is a path

$\Leftarrow$ If Vapor is a path, then $fXg = W$, so W is Sea-Cloud bottleneck



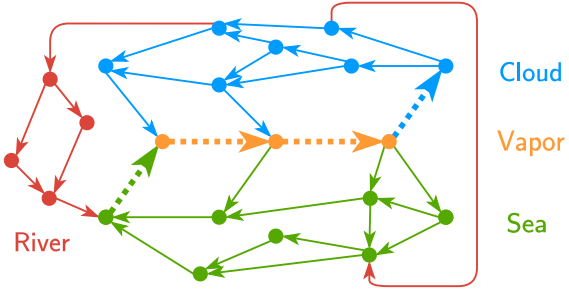
Safety problems as Sea-Cloud connectivity problems

- For any model: if each solution walk goes from Sea to Cloud, then W is safe



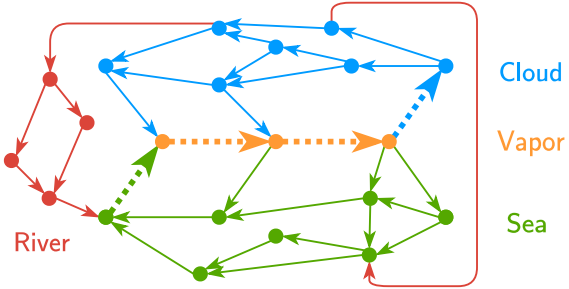
Safety problems as Sea-Cloud connectivity problems

- For any model: if each solution walk goes from Sea to Cloud, then W is safe
- Simplifies single circular



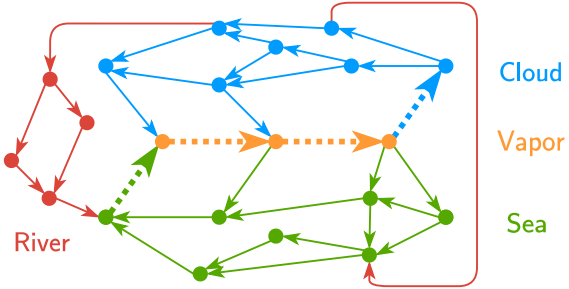
Safety problems as Sea-Cloud connectivity problems

- For any model: if each solution walk goes from Sea to Cloud, then W is safe
- Simplifies single circular
- Simplifies multi circular



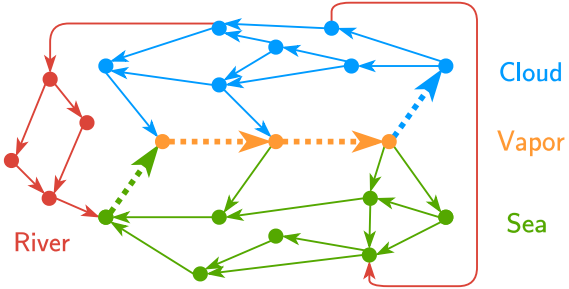
Safety problems as Sea-Cloud connectivity problems

- For any model: if each solution walk goes from Sea to Cloud, then W is safe
- Simplifies single circular
- Simplifies multi circular
- Simplifies single/multi linear



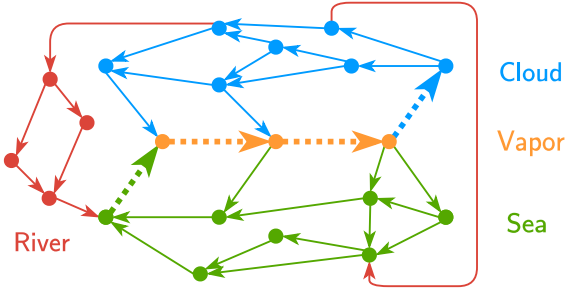
Safety problems as Sea-Cloud connectivity problems

- For any model: if each solution walk goes from Sea to Cloud, then W is safe
- Simplifies single circular
- Simplifies multi circular
- Simplifies single/multi linear
- Simplifies subset covering



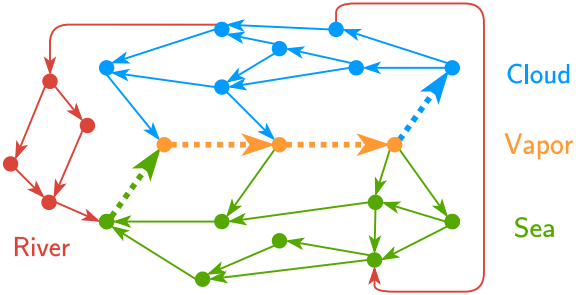
Safety problems as Sea-Cloud connectivity problems

- For any model: if each solution walk goes from Sea to Cloud, then W is safe
- Simplifies single circular
- Simplifies multi circular
- Simplifies single/multi linear
- Simplifies subset covering
- Simplifies . . .



Hydrostructure Algorithms

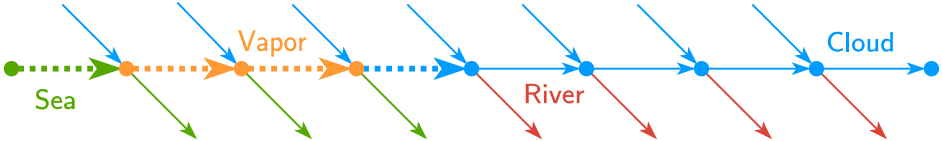
Algorithmic properties: verification



$O(m)$ construction:

- $O(m)$ verification algorithms
  ($O(mn)$ for linear $2 \leq k \leq O(n)$, subset visibility)

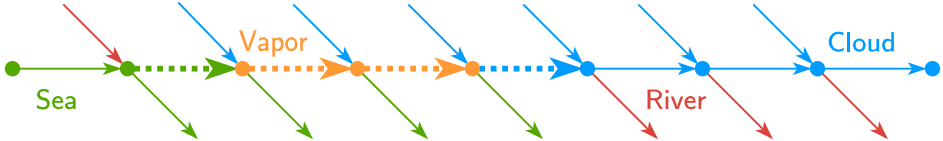
Algorithmic properties: enumeration



$O(m)$ incremental construction:

- $O(m + o)$ enumeration of safe subwalks of a given bottleneck walk ($O(mn)$ for linear $2 \leq k \leq O(n)$, subset visibility)

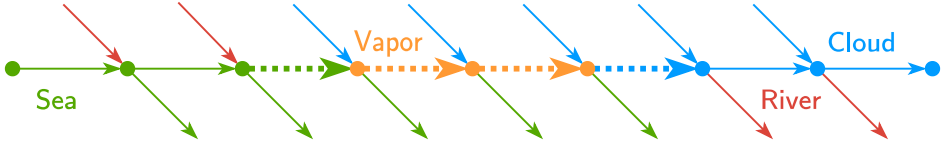
Algorithmic properties: enumeration



***O*(*m*) incremental construction:**

- $O(m + o)$ enumeration of safe subwalks of a given bottleneck walk ($O(mn)$ for linear $2 \leq k \leq O(n)$, subset visibility)

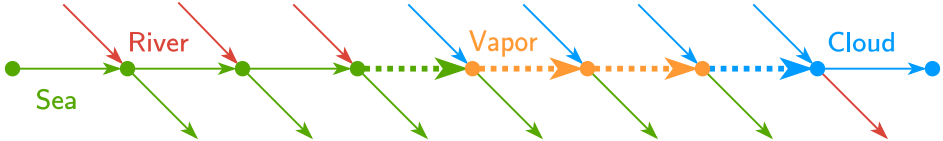
Algorithmic properties: enumeration



$O(m)$ incremental construction:

- $O(m + o)$ enumeration of safe subwalks of a given bottleneck walk ($O(mn)$ for linear $2 \leq k \leq O(n)$, subset visibility)

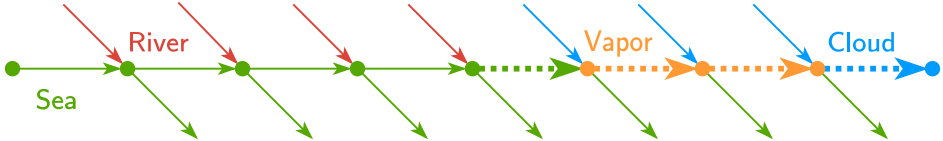
Algorithmic properties: enumeration



$O(m)$ incremental construction:

- $O(m + o)$ enumeration of safe subwalks of a given bottleneck walk ($O(mn)$ for linear $2 \leq k \leq O(n)$, subset visibility)

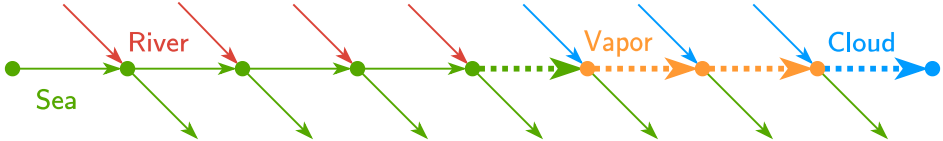
Algorithmic properties: enumeration



***O*(*m*) incremental construction:**

- $O(m + o)$ enumeration of safe subwalks of a given bottleneck walk ($O(mn)$ for linear $2 \leq k \leq O(n)$, subset visibility)

Algorithmic properties: enumeration



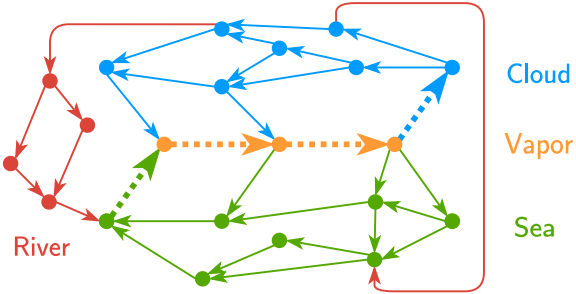
$O(m)$ incremental construction:

- $O(m + o)$ enumeration of safe subwalks of a given bottleneck walk ($O(mn)$ for linear $2 \leq k \leq O(n)$, subset visibility)

Safe walks are subwalks of omnitigs:

- Omnitig enumeration takes $O(mn)$ time and only $O(n)$ of them are bottlenecks
- Optimal $O(mn + o)$ enumeration of all maximal safe walks ($O(m^2 n)$ for linear $2 \leq k \leq O(n)$)

Recap/Conclusion



- The hydrostructure unifies the theory of safe and complete genome assembly
- The hydrostructure yields optimal algorithms for most model combinations

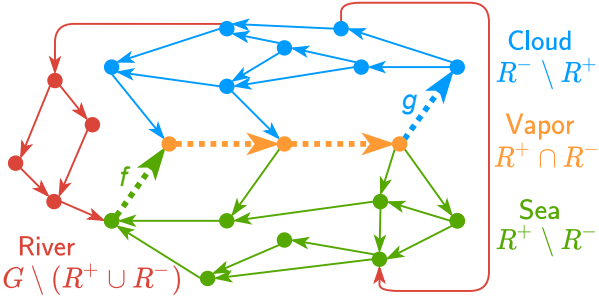
References I

- Acosta, Nidia Obscura, Veli Mäkinen, and Alexandru I. Tomescu (2018). “A safe and complete algorithm for metagenomic assembly”. In: *Algorithms for Molecular Biology* 13.1, 3:1–3:12. DOI: 10.1186/s13015-018-0122-7. URL: https://doi.org/10.1186/s13015-018-0122-7.
- Cairo, Massimo et al. (2019). “An Optimal $O(nm)$ Algorithm for Enumerating All Walks Common to All Closed Edge-covering Walks of a Graph”. In: *ACM Trans. Algorithms* 15.4, 48:1–48:17. DOI: 10.1145/3341731. URL: https://doi.org/10.1145/3341731.
- Cairo, Massimo et al. (2020a). “Genome assembly, a universal theoretical framework: unifying and generalizing the safe and complete algorithms”. In: *arXiv preprint arXiv:2011.12635*.

References II

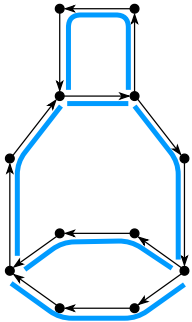
- Cairo, Massimo et al. (2020b). “Genome assembly, from practice to theory: safe, complete and linear-time”. In: *arXiv preprint arXiv:2002.10498.*
- Tomescu, Alexandru I. and Paul Medvedev (2017). “Safe and complete contig assembly through omnitigs”. In: *Journal of Computational Biology* 24.6. Preliminary version appeared in RECOMB 2016., pp. 590–602.

Thank you for attending! Questions?



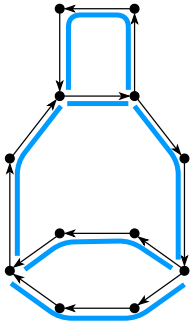
- The hydrostructure unifies the theory of safe and complete genome assembly
- The hydrostructure yields optimal algorithms for most model combinations

Safety in the single circular model

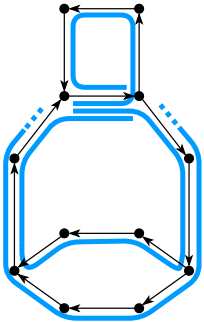


Unitigs

Safety in the single circular model

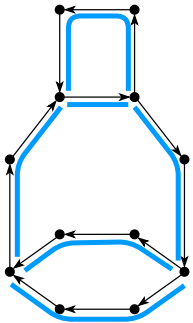


Unitigs

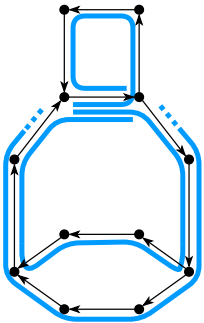


Y-to-V Unitigs
(excerpt)

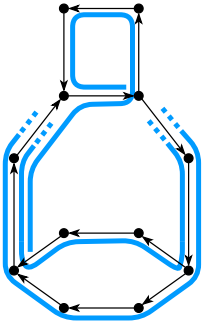
Safety in the single circular model



Unitigs



Y-to-V Unitigs (excerpt)

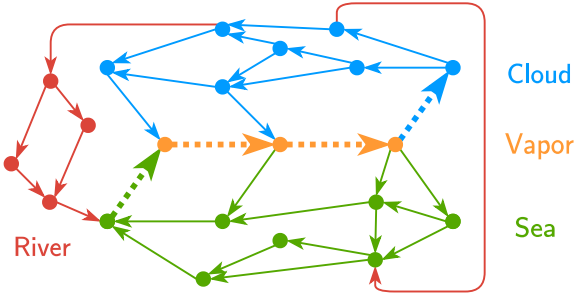


Omnitigs (excerpt) (Tomescu and Medvedev, 2017)

Algorithmic Properties

Verification:

- $O(m)$ verification algorithms ($O(mn)$ for linear $2 \leq k \leq O(n)$ subset visibility)



Algorithmic Properties

Verification:

- $O(m)$ verification algorithms ($O(mn)$ for linear $2 \leq k \leq O(n)$ subset visibility)

Enumeration:

- Optimal $O(mn + o)$ enumeration of all maximal safe walks ($O(m^2 n)$ for linear $2 \leq k \leq O(n)$)