Safety and Completeness in Flow Decompositions for RNA Assembly

Shahbaz Khan, Milla Kortelainen, Manuel Cáceres,
Lucia Williams and Alexandru I. Tomescu

May 2022, RECOMB

Problem & Motivation

Flow decomposition (FD)

Given

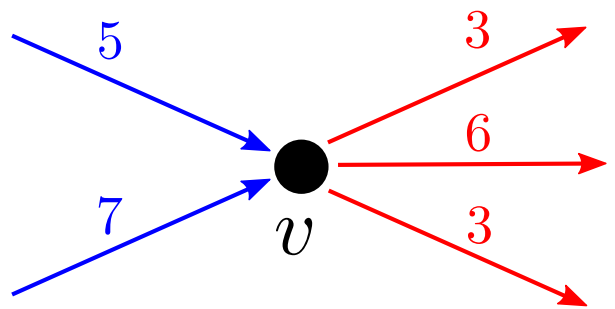
- A directed acyclic graph (DAG) $G = (V, E)$
- An st-flow f on E

Flow decomposition (FD)

Given

- A directed acyclic graph (DAG) $G = (V, E)$
- An st-flow f on E, that is:

$$\forall v \in V \setminus \{s, t\}, \sum_{(u,v)\in E} f(u, v) = f_{in}(v) = f_{out} = \sum_{(v,w)\in E} f(v, w)$$



$$f_{in}(v) = 12 = f_{out}(v)$$

Flow decomposition (FD)

Given

- A DAG $G = (V, E)$, and an st-flow f on E

Report

- A set of st-paths $P_1, \ldots, P_k$, and associated weights $w_1, \ldots, w_k$

decomposing f,

Flow decomposition (FD)

Given

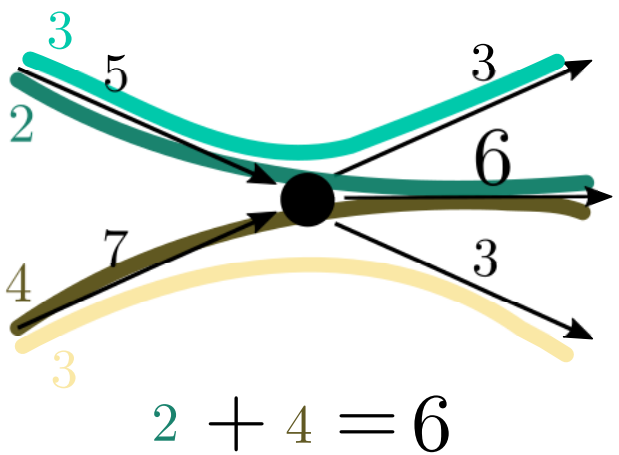
- A DAG $G = (V, E)$, and an st-flow f on E

Report

- A set of st-paths $P_1, \ldots, P_k$, and associated weights $w_1, \ldots, w_k$

decomposing f, that is

$$\forall e \in E, f(e) = \sum_{i, e \in P_i} w_i$$



Applications of FD

- Network routing [13, 8, 12, 23]
- Transportation [24, 25]

Applications of FD

- Network routing [13, 8, 12, 23]
- Transportation [24, 25]
- **Reconstruction of biological sequences: Multi-assembly**
  - **RNA transcript assembly** [27, 33, 10, 6, 32, 37]
  - **Viral quasi-species** [3, 2]

Applications of FD

- Network routing [13, 8, 12, 23]
- Transportation [24, 25]
- **Reconstruction of biological sequences: Multi-assembly**
  - **RNA transcript assembly** [27, 33, 10, 6, 32, 37]
  - **Viral quasi-species** [3, 2]

The flow represents a mixed sample of genomic sequences with different abundances.

Applications of FD

- Network routing [13, 8, 12, 23]
- Transportation [24, 25]
- **Reconstruction of biological sequences: Multi-assembly**
  - **RNA transcript assembly** [27, 33, 10, 6, 32, 37]
  - **Viral quasi-species** [3, 2]

The flow represents a mixed sample of genomic sequences with different abundances.

$\Rightarrow$ A decomposition of that flow tells the different sequences (and their abundances) apart.

Multi-assembly?

The solution is ...

Multi-assembly?

The solution is ...

- A *path cover* [34, 19] (only considers DAG topology)

Multi-assembly?

The solution is ...

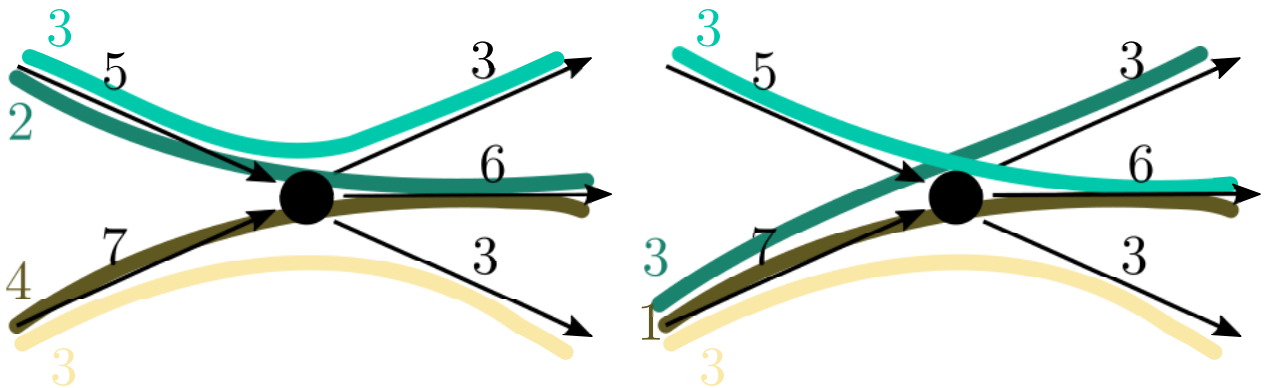
- A *path cover* [34, 19] (only considers DAG topology)
- Some FD
  - Solvable in $O(m(n+m))$ [1] ($m = |E|, n = |V|$)

Multi-assembly?

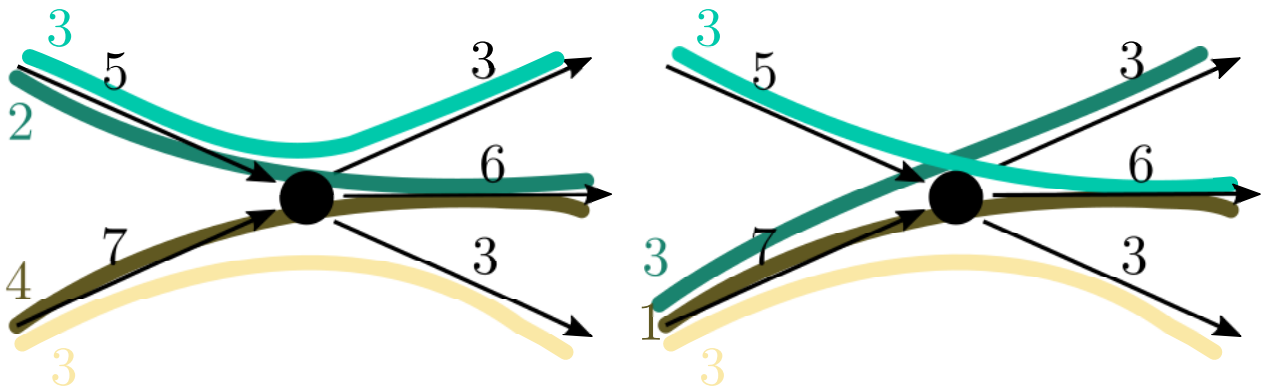
The solution is ...

- A *path cover* [34, 19] (only considers DAG topology)
- Some FD
  - Solvable in $O(m(n+m))$ [1] ($m = |E|, n = |V|$)
- Minimize number of paths (MFD)
  - NP-hard [35], approximations [12, 31, 29, 23, 4, 5], FPT [17], ILP [9], heuristics [33, 27, 30, 17]

Multiple solutions!

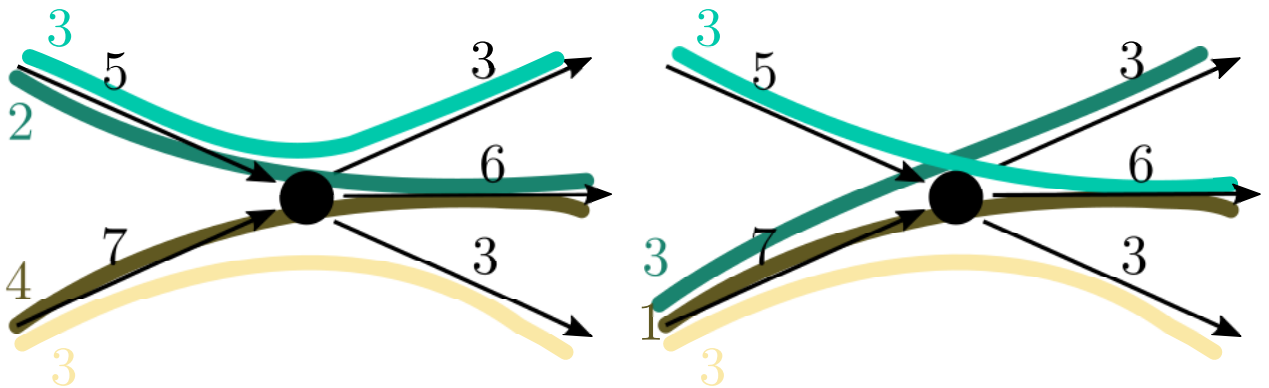


Multiple solutions!



Even in MFD [38]

Multiple solutions!



Even in MFD [38]

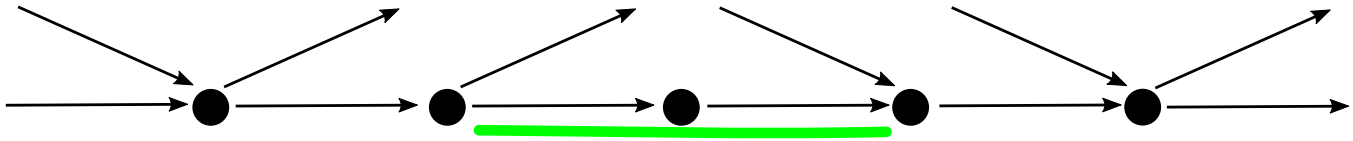
Which is the correct solution?

The Safe Approach

Only report sub-solutions **common to all** solutions

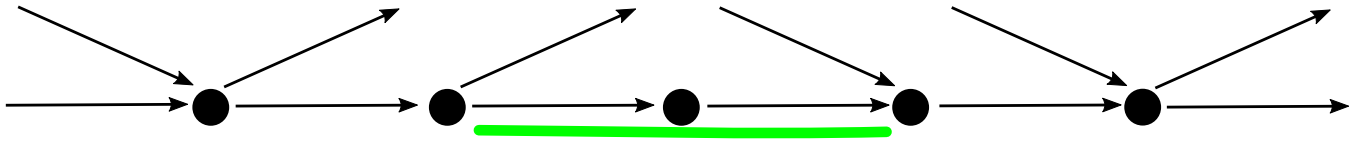
Safe algorithms

- Unitigs [15] in $O(n + m)$

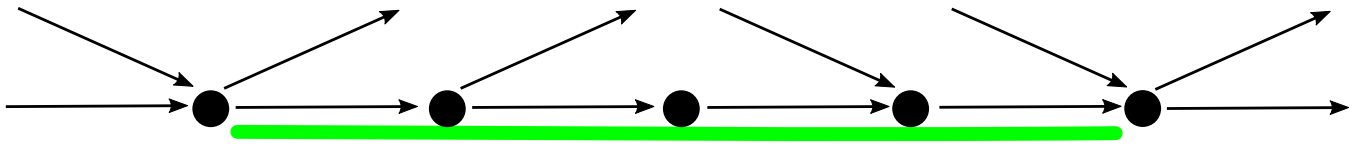


Safe algorithms

- Unitigs [15] in $O(n + m)$

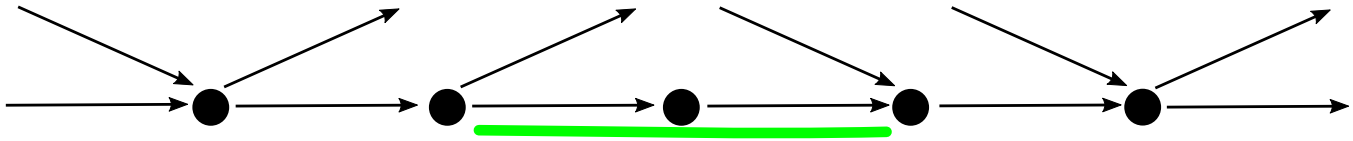


- ExtUnitigs [28, 22, 14, 16] in $O(n + m)$

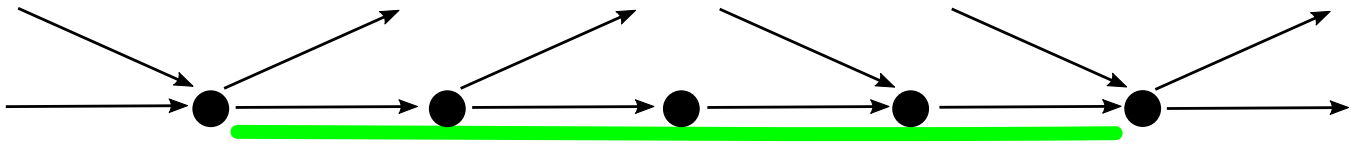


Safe algorithms

- Unitigs [15] in $O(n+m)$



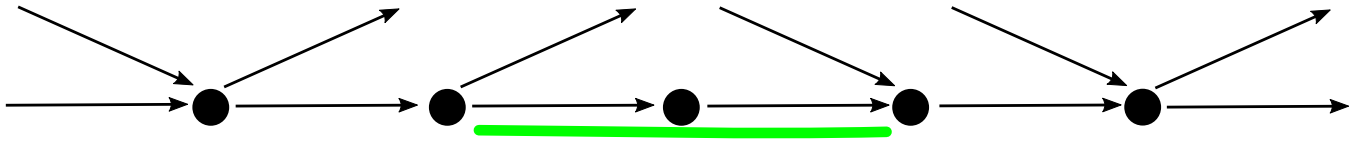
- ExtUnitigs [28, 22, 14, 16] in $O(n+m)$



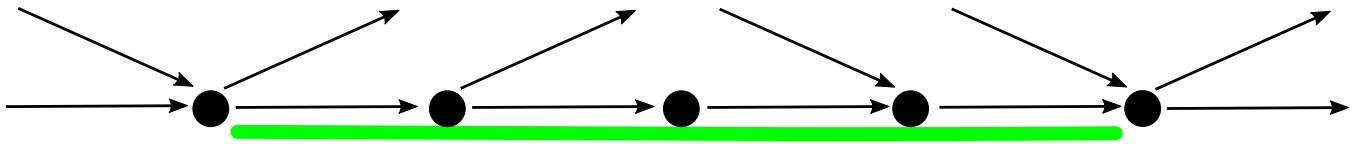
- Safe for covers [7] in $O(k^2nm)$ (k = min. size cover/decomp.)

Safe algorithms

- Unitigs [15] in $O(n+m)$



- ExtUnitigs [28, 22, 14, 16] in $O(n+m)$



- Safe for covers [7] in $O(k^2nm)$ (k = min. size cover/decomp.)

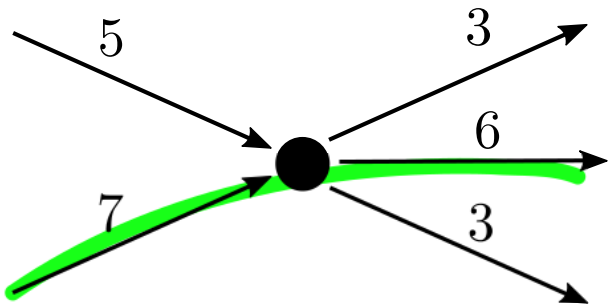
Are these **complete**?

... the flow?

The previous approaches ignore the flow...

... the flow?

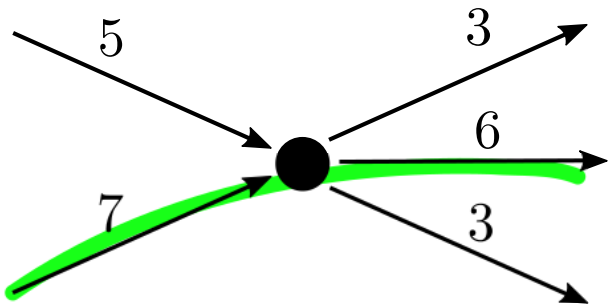
The previous approaches ignore the flow...



One unit of flow must traverse the path on any decomposition

... the flow?

The previous approaches ignore the flow...



One unit of flow must traverse the path on any decomposition

Can we be **Safe&Complete**?

AND-Quant

Recently Ma et al. [20, 21] proposed a method to answer whether any set of edges is safe for FD.

- **Quadratic** algorithm
- Based on a **global** criterion

Our results

Our results

We...

Our results

We...

- propose a **simple and local** characterization of safe paths
- ...leading to

Our results

We...

- propose a **simple and local** characterization of safe paths
- ...leading to
  - a **linear** algorithm to answer if a path is safe
  - a quadratic **Safe&Complete** algorithm

Our results

We...

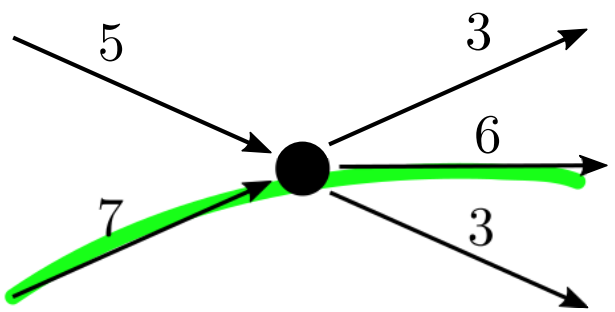
- propose a **simple and local** characterization of safe paths
- ...leading to
  - a **linear** algorithm to answer if a path is safe
  - a quadratic **Safe&Complete** algorithm
- empirically show the advantages of **Safe&Complete** paths in RNA transcript assembly

Theoretical results

Safe paths

Definition (Safe path)

A path P is safe if and only if for every decomposition $P_1, \ldots, P_k$, P is subpath of some P_i

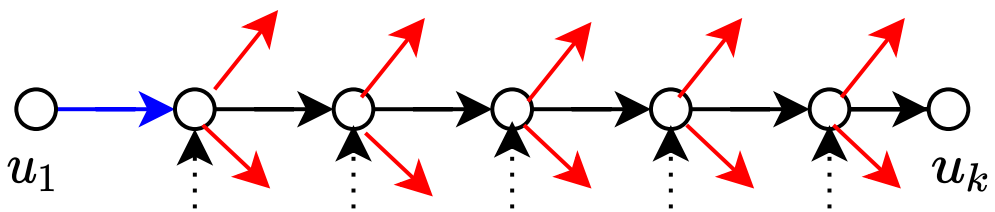


Excess flow

Definition (Excess flow - diverging)

The excess flow f_P of a path $P = u_1, \ldots, u_k$ is

$$f_P = f(u_1, u_2) - \sum_{\substack{u_i \in \{u_2, \ldots, u_{k-1}\} \\ v \neq u_{i+1}}} f(u_i, v)$$

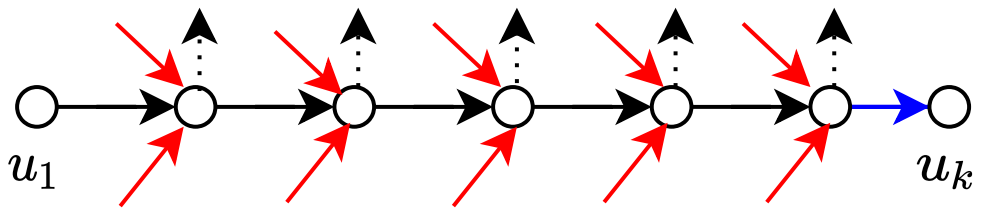


Excess flow

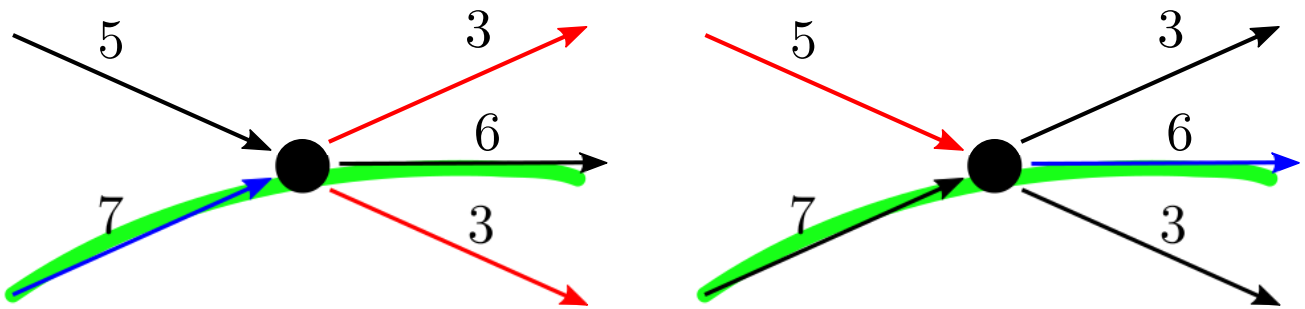
Definition (Excess flow - converging)

The excess flow f_P of a path $P = u_1, \ldots, u_k$ is

$$f_P = f(u_{k-1}, u_k) - \sum_{\substack{u_i \in \{u_2, \ldots, u_{k-1}\} \\ v \neq u_{i-1}}} f(v, u_i)$$



Excess flow – Example



$$f_P = 7 - 3 - 3 = 6 - 5 = 1$$

Excess flow – Efficient

Lemma

Let $P = u_1, \ldots, u_k$ *be a path and* $pP = u_1, \ldots, u_{k-1}$, $sP = u_2, \ldots, u_k$, *then*

$$f_P = f_{pP} + f_{out}(u_2) - f(u_1, u_2) \qquad = f_{in}(u_{k-1}) - f(u_{k-1}, u_k)$$

Excess flow – Efficient

Lemma

Let $P = u_1, \ldots, u_k$ be a path and $pP = u_1, \ldots, u_{k-1}$, $sP = u_2, \ldots, u_k$, then

$$f_P = f_{pP} + f_{out}(u_2) - f(u_1, u_2) \qquad = f_{in}(u_{k-1}) - f(u_{k-1}, u_k)$$

By precomputing $f_{in} = f_{out}$ we obtain

Lemma

We can preprocess G in $O(n+m)$ to compute f_P in $O(|P|)$

Characterization

Theorem

A path P is safe iff its excess flow $f_P > 0$

Characterization

Theorem

A path P is safe iff its excess flow $f_P > 0$

therefore

Theorem

We can preprocess G in $O(n+m)$ to decide if P is safe in $O(|P|)$

Safe&Complete algorithm

1. Precompute the $f_{in} = f_{out}$ values, and a flow decomposition $P_1, \ldots, P_k$
2. For every path P_i run a *two-pointer* algorithm computing the excess flow of subpaths, and reporting maximal safe paths

Practical results

RNA transcript assembly

Datasets

- **Catfish** [30]
  - 100 simulated transcriptomes for human, mouse, and zebrafish using Flux-Simulator [11]
  - 1000 experiments from the Sequence Read Archive, with simulated abundances for transcripts using Salmon [26]
  - Small number of *complex instances* (large k)
- **Reference-Sim** [36]
  - For each transcript in the GRCh.104 *homo sapiens* reference genome, it samples a value from a lognormal distribution using RNASeqReadSimulator [18]
  - Larger number of *complex instances*

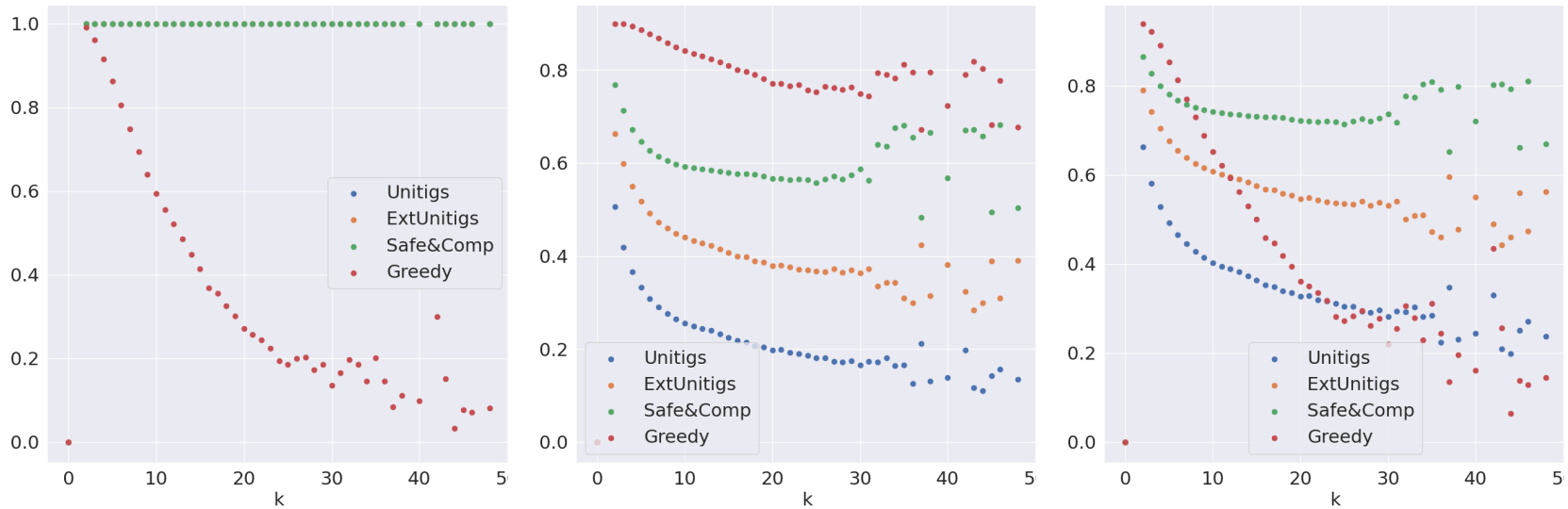
Evaluation metrics

Weighted precision: Total length of correctly reported paths divided by the total length of reported paths.

Maximum relative coverage: Length of the longest segment of a reported path inside a transcript T, divided by $|T|$

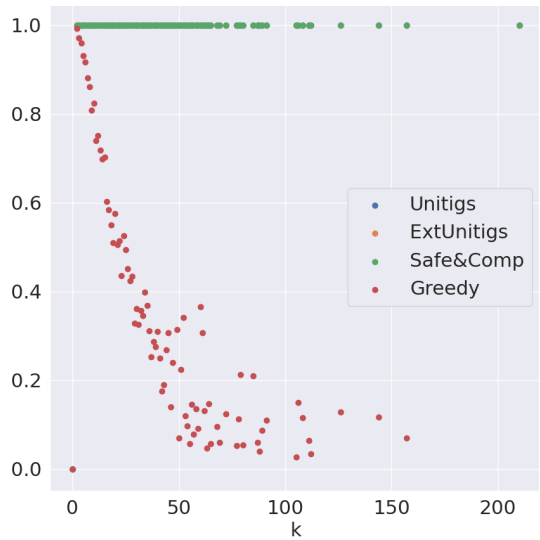
F-score: Harmonic mean of weighted precision and maximum relative coverage

Results – Catfish

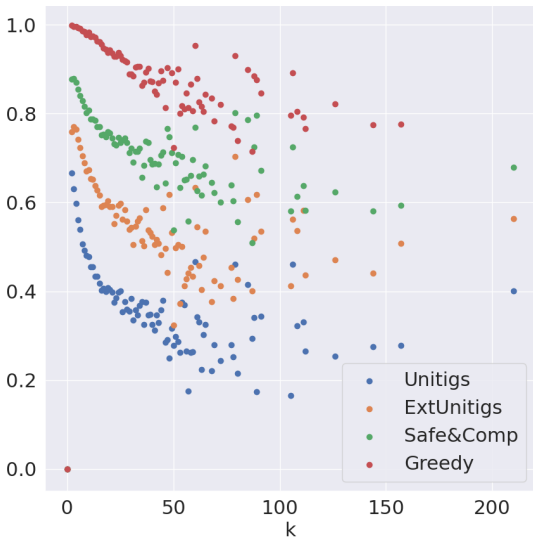


(a) Weighted Precision (b) Max. Rel. Coverage (c) F-Score

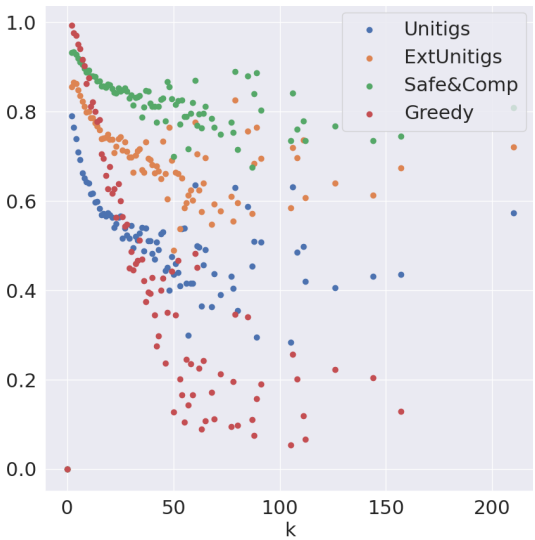
Results – Reference-Sim



(a) Weighted Precision



(b) Max. Rel. Coverage



(c) F-Score

Results – Performance

Algorithm	Reference-Sim		Catfish			
	Human 25.6MB		Zebrafish 122MB		Human (salmon) 2.5GB	
	Time	Mem	Time	Mem	Time	Mem
Unitigs	0.68s	3.58MB	13.82s	3.51MB	303.72s	3.66MB
ExtUnitigs	0.99s	3.63MB	18.31s	3.52MB	404.50s	3.68MB
Safe&Comp	2.56s	4.47MB	20.17s	3.56MB	667.27s	3.84MB
Greedy	7.71s	4.88MB	108.30s	6.00MB	2684.30s	8.47MB

Funding


erc
European Research Council
Established by the European Commission


NSF

References I

[1] Ahuja, R. K., Magnanti, T. L., and Orlin, J. B.
Network flows - theory, algorithms and applications.
Prentice Hall, 1993.

[2] Baaijens, J. A., der Roest, B. V., Köster, J., Stougie, L., and Schönhuth, A.
Full-length de novo viral quasispecies assembly through variation graph construction.
Bioinform. 35, 24 (2019), 5086–5094.

[3] Baaijens, J. A., Stougie, L., and Schönhuth, A.
Strain-aware assembly of genomes from mixed samples using flow variation graphs.
In *Research in Computational Molecular Biology - 24th Annual International Conference, RECOMB 2020, Padua, Italy, May 10-13, 2020, Proceedings* (2020), R. Schwartz,

References II

Ed., vol. 12074 of *Lecture Notes in Computer Science*, Springer, pp. 221–222.

[4] Baier, G., Köhler, E., and Skutella, M.
On the k-splittable flow problem.
In *European Symposium on Algorithms* (2002), Springer, pp. 101–113.

[5] Baier, G., Köhler, E., and Skutella, M.
The k-splittable flow problem.
Algorithmica 42, 3-4 (2005), 231–248.

[6] Bernard, E., Jacob, L., Mairal, J., and Vert, J.-P.
Flipflop: Fast lasso-based isoform prediction as a flow problem, 2013.

References III

[7] Caceres, M., Mumey, B., Husic, E., Rizzi, R., Cairo, M., Sahlin, K., and Tomescu, A. I. I.
Safety in multi-assembly via paths appearing in all path covers of a DAG.
IEEE/ACM Transactions on Computational Biology and Bioinformatics (2021).

[8] Cohen, R., Lewin-Eytan, L., Naor, J. S., and Raz, D.
On the effect of forwarding table size on sdn network utilization.
In *IEEE INFOCOM 2014-IEEE conference on computer communications* (2014), IEEE, pp. 1734–1742.

References IV

[9] Dias, F. H. C., Williams, L., Mumey, B., and Tomescu, A. I.
Fast, Flexible, and Exact Minimum Flow Decompositions via ILP.
arXiv arXiv:2201.10923 (2022).
To appear in the Proceedings of RECOMB 2022 – 26th Annual International Conference on Research in Computational Molecular Biology.

[10] Gatter, T., and Stadler, P. F.
Ryūtō: network-flow based transcriptome reconstruction.
BMC bioinformatics 20, 1 (2019), 190.

References V

[11] Griebel, T., Zacher, B., Ribeca, P., Raineri, E., Lacroix, V., Guigó, R., and Sammeth, M.
Modelling and simulating generic rna-seq experiments with the flux simulator.
Nucleic acids research 40, 20 (2012), 10073–10083.

[12] Hartman, T., Hassidim, A., Kaplan, H., Raz, D., and Segalov, M.
How to split a flow?
In *2012 Proceedings IEEE INFOCOM* (2012), IEEE, pp. 828–836.

[13] Hong, C.-Y., Kandula, S., Mahajan, R., Zhang, M., Gill, V., Nanduri, M., and Wattenhofer, R.
Achieving high utilization with software-driven wan.
In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM* (2013), pp. 15–26.

References VI

[14] Jackson, B. G.
Parallel methods for short read assembly.
PhD thesis, Iowa State University, 2009.

[15] Kececioglu, J. D., and Myers, E. W.
Combinatorial algorithms for DNA sequence assembly.
Algorithmica 13, 1/2 (1995), 7–51.

[16] Kingsford, C., Schatz, M. C., and Pop, M.
Assembly complexity of prokaryotic genomes using short reads.
BMC Bioinformatics 11, 1 (2010), 21.

References VII

[17] Kloster, K., Kuinke, P., O'Brien, M. P., Reidl, F., Villaamil, F. S., Sullivan, B. D., and van der Poel, A.
A practical fpt algorithm for flow decomposition and transcript assembly.
In *2018 Proceedings of the Twentieth Workshop on Algorithm Engineering and Experiments (ALENEX)* (2018), SIAM, pp. 75–86.

[18] Li, W.
RNASeqReadSimulator: a simple RNA-seq read simulator, 2014.

[19] Liu, R., and Dickerson, J.
Strawberry: Fast and accurate genome-guided transcript reconstruction and quantification from rna-seq.
PLoS computational biology 13, 11 (2017), e1005851.

References VIII

[20] Ma, C., Zheng, H., and Kingsford, C.
Exact transcript quantification over splice graphs.
In *20th International Workshop on Algorithms in Bioinformatics, WABI 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference)* (2020), C. Kingsford and N. Pisanti, Eds., vol. 172 of *LIPIcs*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 12:1–12:18.

[21] Ma, C., Zheng, H., and Kingsford, C.
Finding ranges of optimal transcript expression quantification in cases of non-identifiability.
bioRxiv (2020).
To appear at RECOMB 2021.

References IX

[22] Medvedev, P., Georgiou, K., Myers, G., and Brudno, M.
Computability of models for sequence assembly.
In *WABI* (2007), pp. 289–301.

[23] Mumey, B., Shahmohammadi, S., McManus, K., and Yaw, S.
Parity balancing path flow decomposition and routing.
In *2015 IEEE Globecom Workshops (GC Wkshps)* (2015), IEEE, pp. 1–6.

[24] Ohst, J. P.
On the Construction of Optimal Paths from Flows and the Analysis of Evacuation Scenarios.
PhD thesis, University of Koblenz and Landau, Germany, 2015.

References X

[25] Olsen, N., Kliewer, N., and Wolbeck, L.
A study on flow decomposition methods for scheduling of electric buses in public transport based on aggregated time–space network models.
Central European Journal of Operations Research (2020).

[26] Patro, R., Duggal, G., and Kingsford, C.
Salmon: accurate, versatile and ultrafast quantification from rna-seq data using lightweight-alignment.
BioRxiv (2015), 021592.

[27] Pertea, M., Pertea, G. M., Antonescu, C. M., Chang, T.-C., Mendell, J. T., and Salzberg, S. L.
Stringtie enables improved reconstruction of a transcriptome from rna-seq reads.
Nature biotechnology 33, 3 (2015), 290–295.

References XI

[28] Pevzner, P. A., Tang, H., and Waterman, M. S.
An Eulerian path approach to DNA fragment assembly.
Proceedings of the National Academy of Sciences 98, 17 (2001), 9748–9753.

[29] Pieńkosz, K., and Kołtyś, K.
Integral flow decomposition with minimum longest path length.
European Journal of Operational Research 247, 2 (2015), 414–420.

[30] Shao, M., and Kingsford, C.
Theory and a heuristic for the minimum path flow decomposition problem.
IEEE/ACM Transactions on Computational Biology and Bioinformatics 16, 2 (2017), 658–670.

References XII

[31] Suppakitpaisarn, V.
An approximation algorithm for multiroute flow decomposition.
Electronic Notes in Discrete Mathematics 52 (2016), 367 – 374.
INOC 2015 – 7th International Network Optimization Conference.

[32] Tomescu, A. I., Gagie, T., Popa, A., Rizzi, R., Kuosmanen, A., and Mäkinen, V.
Explaining a weighted DAG with few paths for solving genome-guided multi-assembly.
IEEE ACM Trans. Comput. Biol. Bioinform. 12, 6 (2015), 1345–1354.

References XIII

[33] Tomescu, A. I., Kuosmanen, A., Rizzi, R., and Mäkinen, V.
A novel min-cost flow method for estimating transcript expression with rna-seq.
BMC bioinformatics 14, S5 (2013), S15.

[34] Trapnell, C., Williams, B. A., Pertea, G., Mortazavi, A., Kwan, G., Van Baren, M. J., Salzberg, S. L., Wold, B. J., and Pachter, L.
Transcript assembly and quantification by rna-seq reveals unannotated transcripts and isoform switching during cell differentiation.
Nature biotechnology 28, 5 (2010), 511–515.

References XIV

[35] Vatinlen, B., Chauvet, F., Chrétienne, P., and Mahey, P.
Simple bounds and greedy algorithms for decomposing a flow into a minimal set of paths.
European Journal of Operational Research 185, 3 (2008), 1390–1401.

[36] Williams, L.
Reference-sim, Nov. 2021.

[37] Williams, L., Reynolds, G., and Mumey, B.
Rna transcript assembly using inexact flows.
In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (2019), IEEE, pp. 1907–1914.

References XV

[38] Williams, L., Tomescu, A., Mumey, B. M., et al.
Flow decomposition with subpath constraints.
In *21st International Workshop on Algorithms in Bioinformatics (WABI 2021)* (2021), Schloss Dagstuhl-Leibniz-Zentrum für Informatik.