Algorytmy rozproszone i metody obliczeniowe w skalowalnym przetwarzaniu danych pochodzących z sekwencjonowania wysokoprzepustowego

Marek Wiewiórka
promotor: dr hab. inż. Tomasz Gambin, prof. uczelni

Instytut Informatyki, Politechnika Warszawska

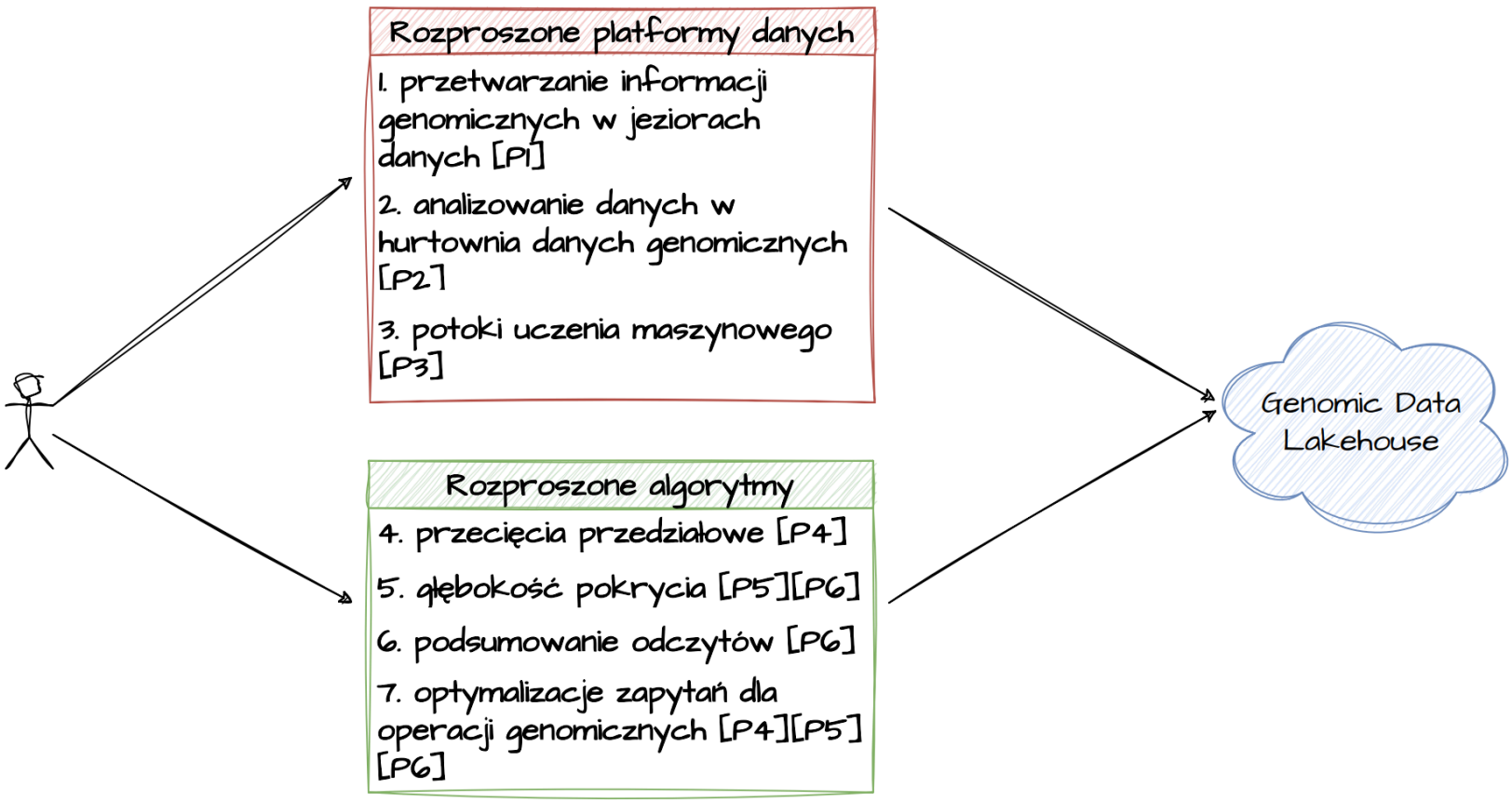
21 września 2023

Konspekt

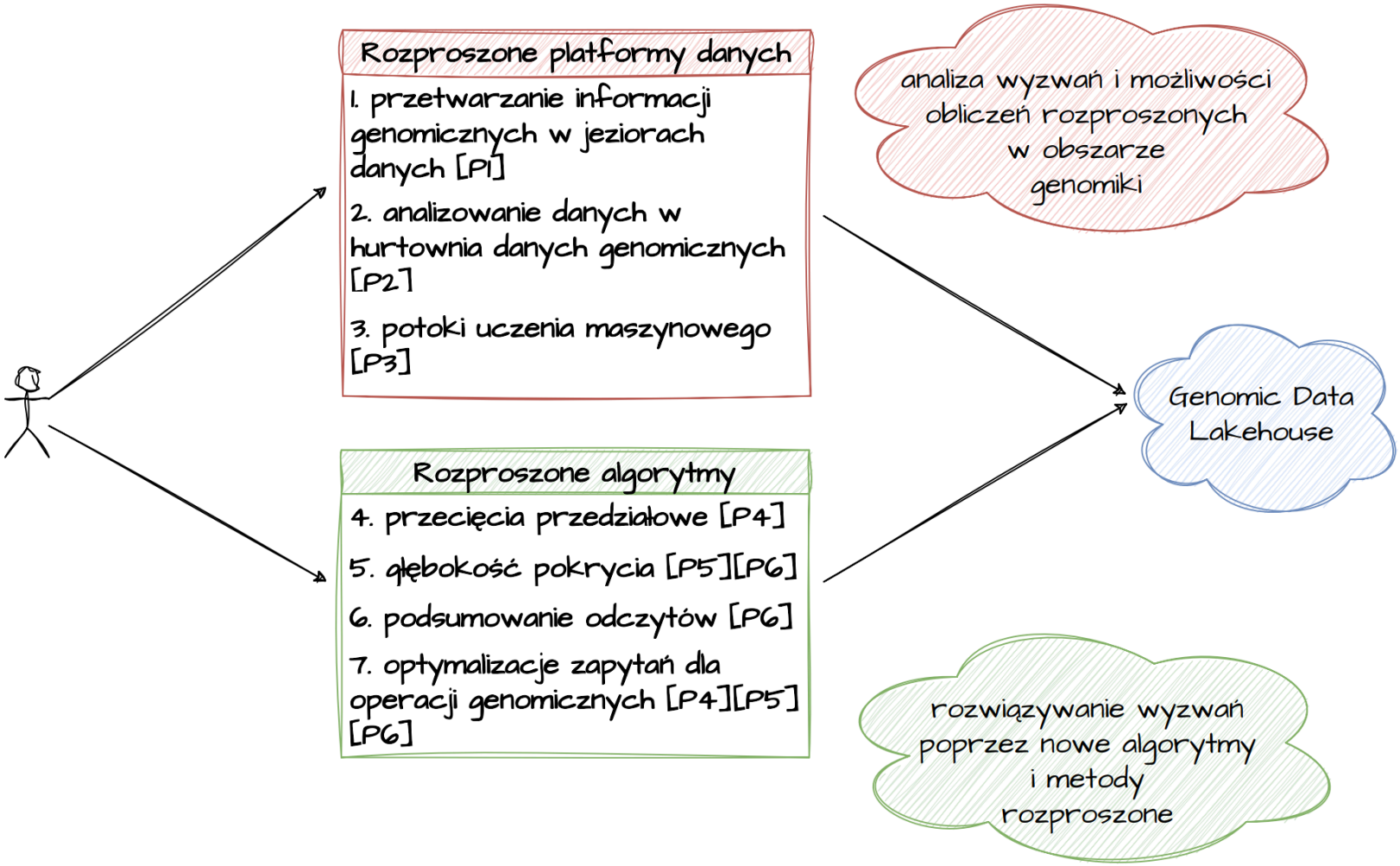
Główne tezy badawcze

1. Zastosowanie **rozproszonych silników obliczeniowych** może znacznie poprawić wydajność analiz genomicznych w środowisku jeziora danych (ang. *Data Lake*).
2. Kolumnowe formaty danych wraz z nowoczesnymi silnikami zapytań SQL i odpowiednimi technikami modelowania danych umożliwiają **zbudowanie rozproszonej platformy hurtowni danych** dla populacyjnych analiz genomicznych.
3. Budowa skalowalnej platformy typu *Data Lakehouse* dla danych genomicznych wymaga **nowych rozproszonych algorytmów** dla typowych operacji bioinformatycznych, takich jak: **przecięcia przedziałów, obliczanie głębokości pokrycia, podsumowanie odczytów**.

Obszary badawcze I



Obszary badawcze II



Publikacje I

[P1] M.S. Wiewiórka, A. Messina, A. Pacholewska, S. Maffioletti, P. Gawrysiak, and M.J. Okoniewski. SparkSeq: Fast, scalable and cloud-ready tool for the interactive genomic data analysis with nucleotide precision.
Bioinformatics, 30(18), 2014, **MNiSW list: 200 pts**, **Impact factor: 4.981**, **Contribution: 55%**

[P2] M.S. Wiewiórka, D.P. Wysakowicz, M.J. Okoniewski, and T. Gambin. Benchmarking distributed data warehouse solutions for storing genomic variant information.
Database : the journal of biological databases and curation, 2017, 2017, **MNiSW list: 100 pts**, **Impact factor: 2.627**, **Contribution: 65%**

[P3] Anastasiia Hryhorzhevska, Marek Wiewiórka, Michał Okoniewski, and Tomasz Gambin. Scalable Framework for the Analysis of Population Structure Using the Next Generation Sequencing Data.
In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 10352 LNAI, pages 471–480. 2017, **MNiSW list: 20 pts**, **Impact factor: –**, **Contribution: 20%**

Publikacje II

[P4] Marek Wiewiórka, Anna Leśniewska, Agnieszka Szmurło, Kacper Stępień,

Mateusz Borowiak, Michał Okoniewski, and Tomasz Gambin. SeQuiLa: an elastic, fast and scalable SQL-oriented solution for processing and querying genomic intervals.
Bioinformatics, 35(12):2156–2158, June 2019, **MNiSW list: 200 pts**, **Impact factor: 4.531**, **Contribution: 51%**

[P5] Marek Wiewiórka, Agnieszka Szmurło, Wiktor Kuśmirek, and Tomasz

Gambin. SeQuiLa-cov: A fast and scalable library for depth of coverage calculations.
GigaScience, 8(8), August 2019, **MNiSW list: 200 pts**, **Impact factor: 5.71**, **Contribution: 45%**

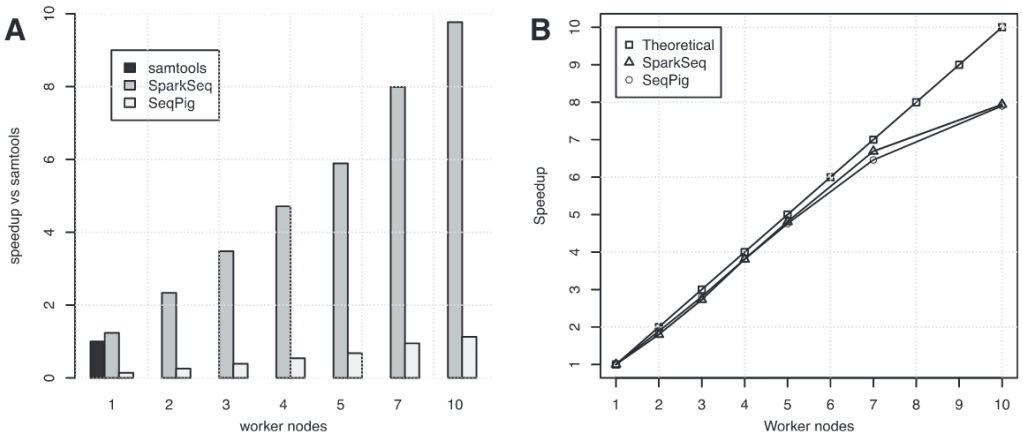
[P6] Marek Wiewiórka, Agnieszka Szmurło, Paweł Stankiewicz, and Tomasz

Gambin. Cloud-native distributed genomic pileup operations.
Bioinformatics, December 2022, **MNiSW list: 200 pts**, **Impact factor: 6.931**, **Contribution: 45%**

Rozproszone platformy danych

Przetwarzanie informacji genomicznej w jeziorach danych [P1]

- **[P1]** jako jedna z **pierwszych** publikacji na świecie dotycząca zastosowana rozproszonej platformy programistycznej Apache Spark do analizy danych genomicznych
- implementacja kilku przykładowych operacji genomicznych przy wykorzystaniu RDD API oraz odczyt formatów danych bioinformatycznych przy pomocy Hadoop-BAM[1]
- porównanie wydajności z narzędziami bazującymi na platformie programistycznej Hadoop **Map-Reduce** – SeqPig[2], np. filtrowanie odczytów wg kilku warunków i wyliczanie statystyk podsumowujących



- łatwość programowania, możliwość pracy z bioinformatycznymi formatami danych oraz bardzo dobra wydajność i skalowalność.

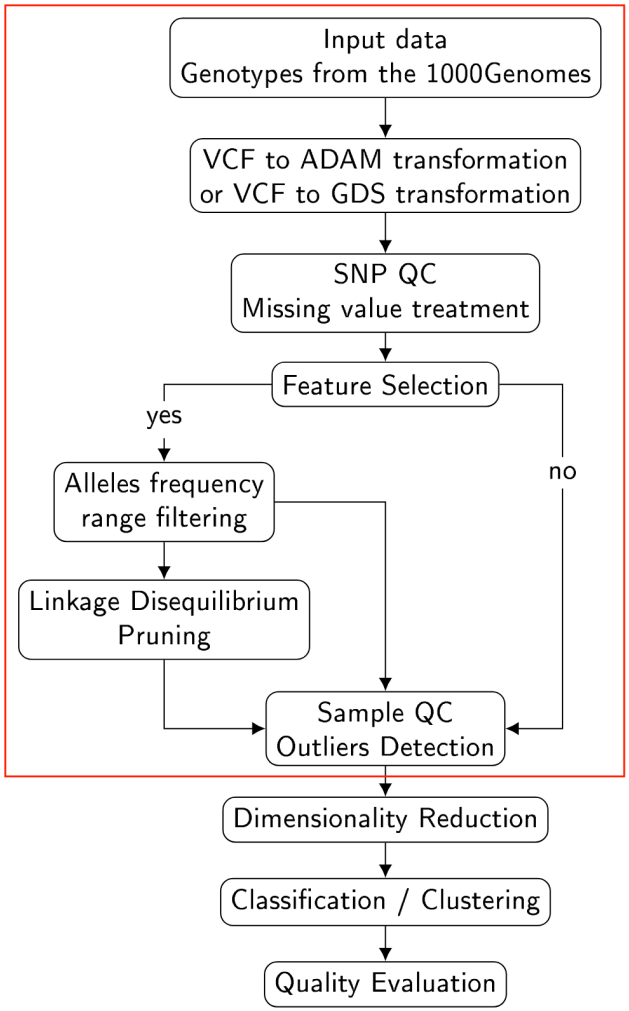
Rozproszone hurtownie danych dla informacji genomicznych [P2]

- w **[P2]** zaproponowana metodologia testowa inspirowana TPC-DS[3] dla genomicznej hurtowni danych składająca się z:
  - podstawowego modelu danych w formacie gwiazdy
  - generatora danych o wariantach na podstawie bazy dbNSFP[4] oraz informacji o ich częstości z bazy ExAC[5]
  - zestawu 12 kwerend
- „***Apache Spark*** *2 wydaje się być najbardziej uniwersalnym narzędziem w badaniu. Jest odpowiednie zarówno do złożonych zapytań, jak i porównywalne z Presto, w przypadku mniej wymagającej, interaktywnej analizy danych*“

Query	Level	Hive on MR [s]		Presto[s]		Spark 1[s]		Spark 2[s]		Impala [s]		MonetDB [s]	Kylin [s]
	Format	ORC	Parquet	ORC	Parquet	ORC	Parquet	ORC	Parquet	Parquet	Kudu	Custom	HFile
$Q1_A$	raw	648	1548	283	314	572	434	330	**280**	814	1523		
	aggr	216	212	25	28	55	35	27	**23**	85	54	207	
	aggr+denorm	123	315	11	19	24	21	15	11	35	83	182	**0.32**
$Q1_B$	raw	710	1603	270	316	230	**145**	335	219	685	1580		
	aggr	193	195	24	28	39	27	25	**22**	84	56	44	
	aggr+denorm	141	367	13	19	21	20	17	14	49	90	143	**0.85**

Rozproszone potoki uczenia maszynowego [P3]

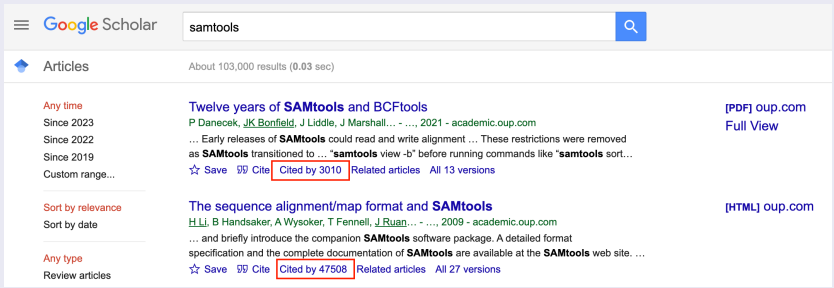
- implementacja pełnego potoku uczenia maszynowego dla danych z projektu **1000 Genomes** przy użyciu platformy programistycznej **Apache Spark**
- potwierdzenie dużej przydatności rozproszonych metod dla zadań związanych z wstępnym przygotowaniem danych oraz **inżynierią cech** – duża liczba **próbek**
- istotne problemy wydajnościowe w przypadku wykorzystania metod **redukcji wymiaru** – duża liczba **cech**
- konieczność stosowanie zoptymalizowanych i łatwych do rozproszenia metod, takich jak np. REGENIE[6]



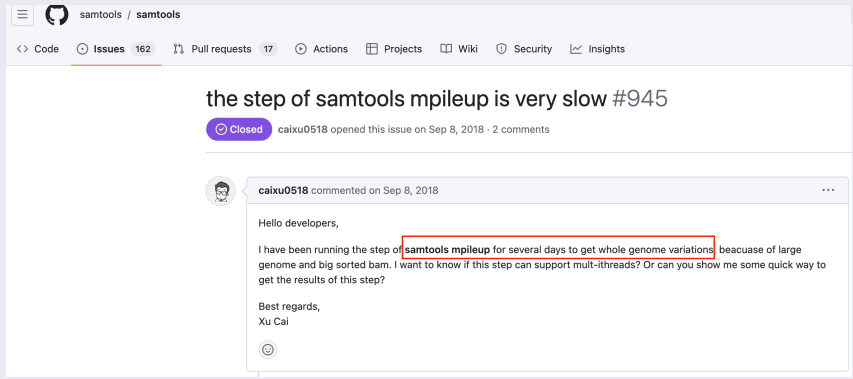
Rozproszone algorytmy dla operacji genomicznych

Podsumowanie odczytów i głębokość pokrycia [P5],[P6]

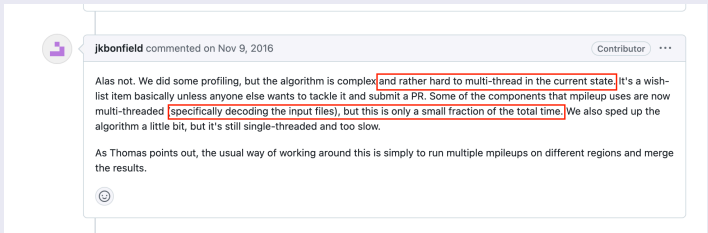
Obie operacje są powszechnie wykorzystywane w wielu analizach, jak np. wyszukiwanie polimorfizmów pojedynczych nukleotydów czy zmienności liczby kopii...



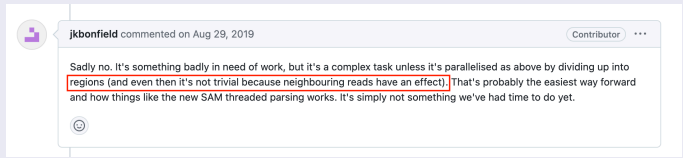
...i są często jednymi z najbardziej czasochłonnych etapów...



...operacje nie są łatwe do zrównoleglenia...

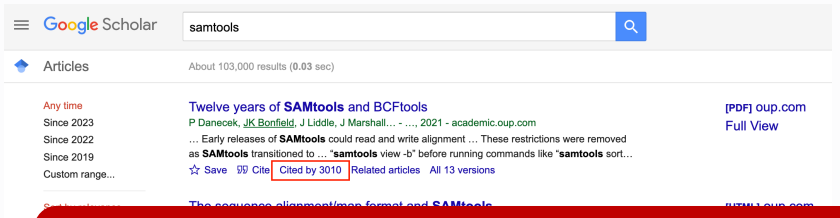


...m.in. ze względu na przynależność odczytów do sąsiadujących regionów...

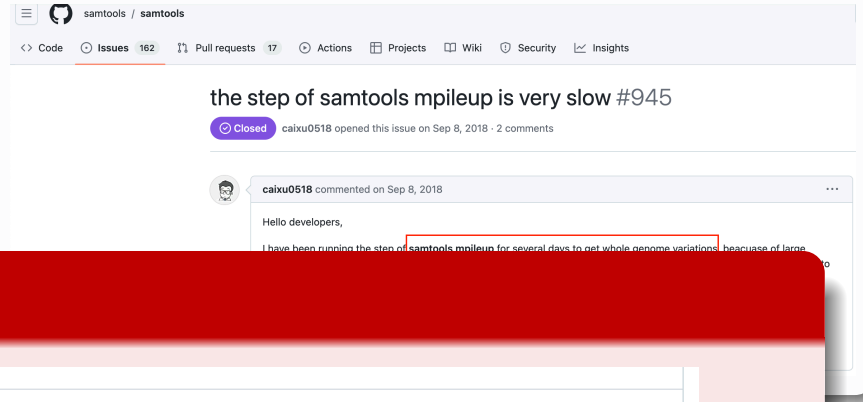


Podsumowanie odczytów i głębokość pokrycia [P5],[P6]

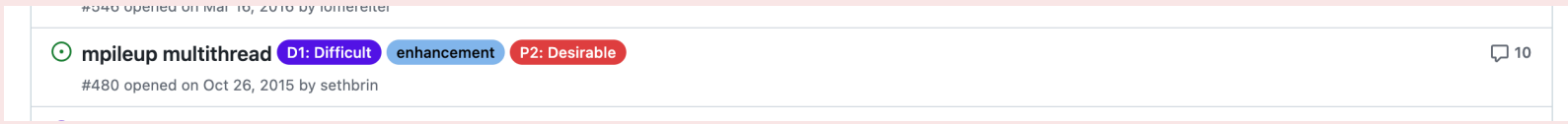
Obie operacje są powszechnie wykorzystywane w wielu analizach, jak np. wyszukiwanie polimorfizmów pojedynczych nukleotydów czy zmienności liczby kopii...



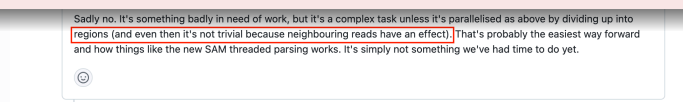
...i są często jednymi z najbardziej czasochłonnych etapów...



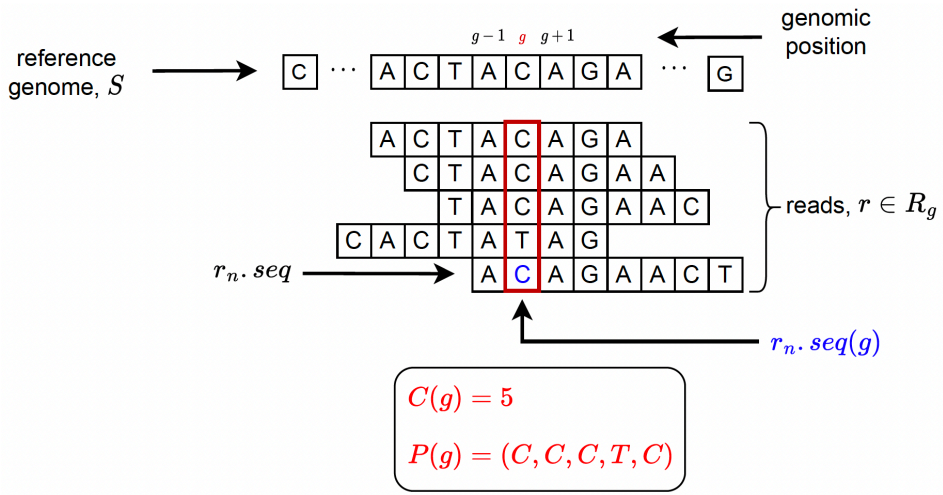
Cel



Opracować **wydajną** metodę rozproszoną.

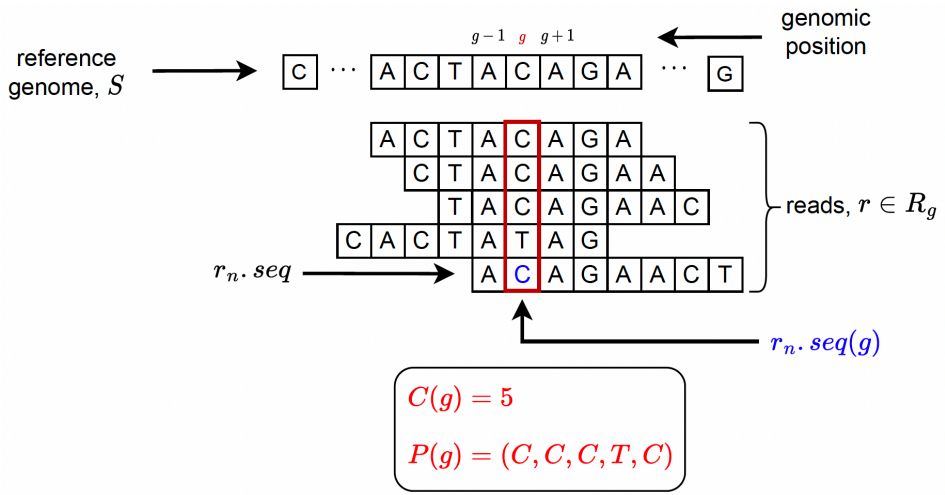


Definicja problemu [P5],[P6]



- konieczność dekodowania ***wszystkich sekwencji odczytów*** z R_g
- algorytmy nie używają informacji o ***jakości dopasowania*** odczytów do sekwencji referencyjnej

Definicja problemu [P5],[P6]



- konieczność dekodowania ***wszystkich sekwencji odczytów*** z R_g
- algorytmy nie używają informacji o ***jakości dopasowania*** odczytów do sekwencji referencyjnej

Sekwencja referencyjna S jest złożona z symboli pochodzących z alfabetu $\Sigma = \{A, C, G, T\}$, G oznacza zbiór wszystkich pozycji genomicznych, a R to zbiór odczytów zmapowanych do sekwencji referencyjnej S, natomiast R_g to podzbiór odczytów pokrywających pozycję genomiczną g , wtedy dla każdej pozycji $g \in G$ możemy zdefiniować dwie funkcje:
podsumowanie odczytów:

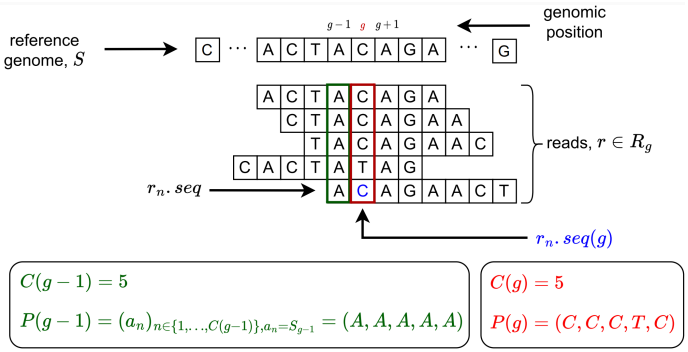
$$P(g) = \{(a_1, \ldots, a_n) | r \in R_g \wedge n \in \{1, \ldots, |R_g|\} \wedge a_n = r_n.seq(g)\} \quad (1)$$

głębokości pokrycia:

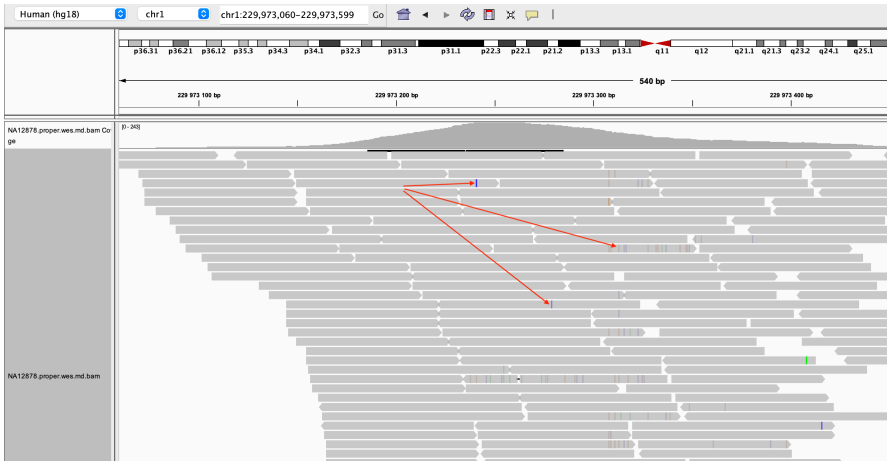
$$C(g) = |P(g)| = |(a_1, \ldots, a_n)| = |R_g| \quad (2)$$

Podejście zoptymalizowane [P5],[P6]

- statystyka rozkładu dopasowania odczytów
- adaptacyjna strategia



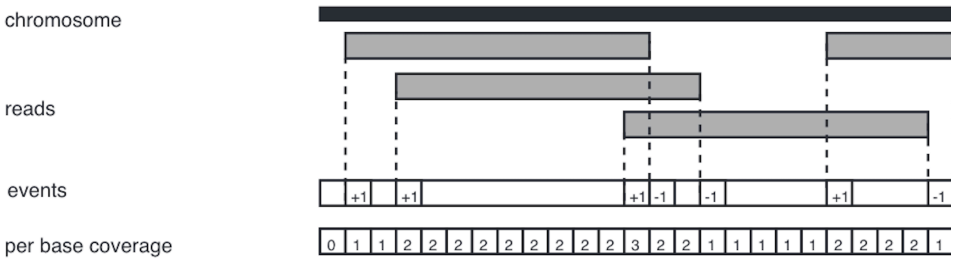
- brak konieczności iterowania sekwencji odczytów



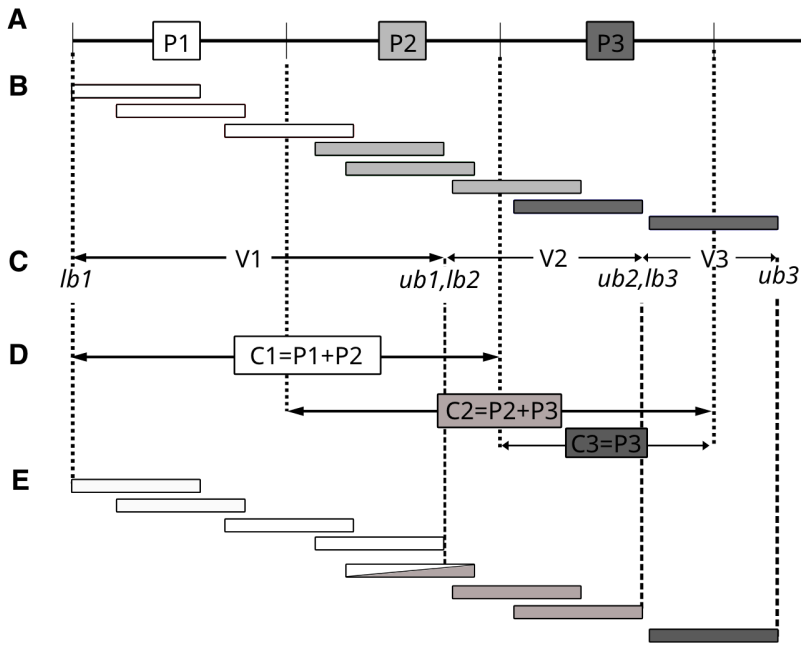
CIGAR	MD tag	Freq.	Cum. Freq.
76M	76	0,683	0,683
101M	101	0,124	**0,807**
76M	0A75	0,001	0,817

Metoda rozproszona [P5],[P6]

- wyznaczenie wirtualnych i scalonych partycji
- wyznaczenie dla każdej wirtualnej partycji struktury zawierającej zdarzenia początku i końca odczytów zgodnie z mosdepth [7]



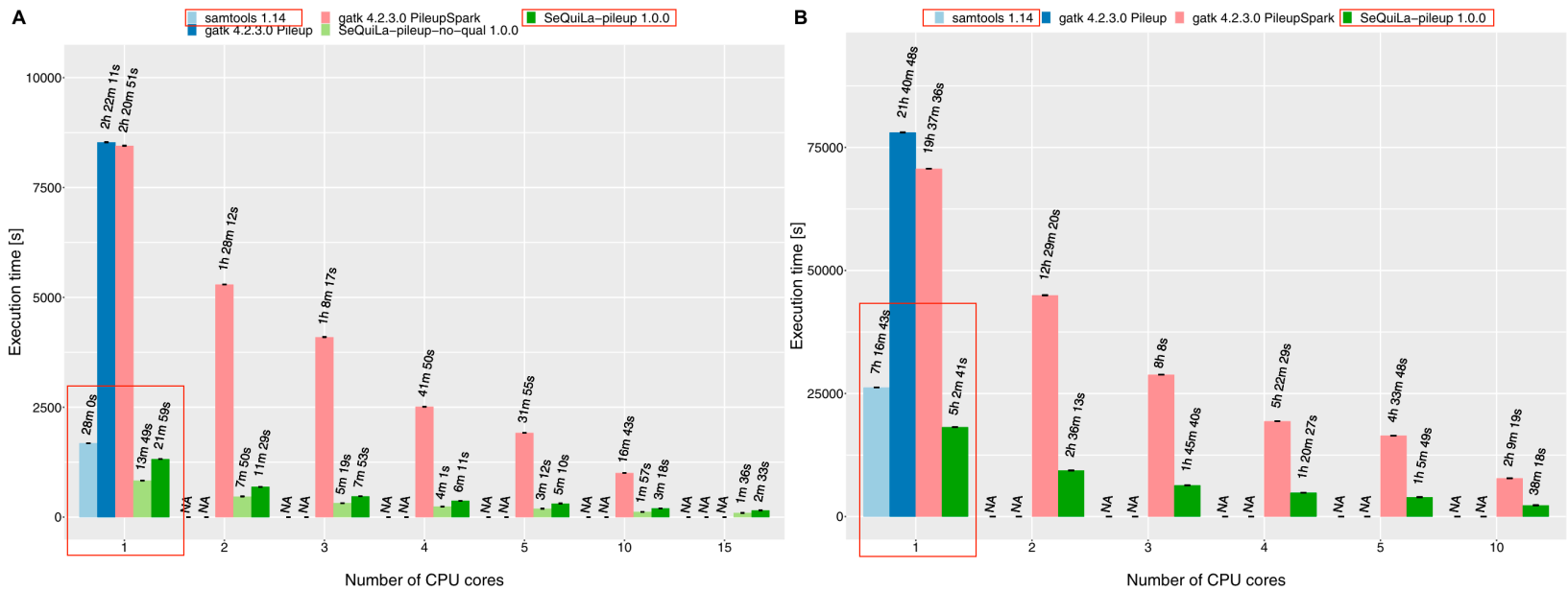
- wyznaczanie struktur dla alternatywnych alleli (oraz opcjonalnie miar jakości)
- generowanie wyników (głębokość pokrycia, podsumowanie odczytów)



Rysunek: (**A**) Domyślne partycje; (**B**) wirtualne partycje do obliczeń; (**C**) scalone partycje do odczytu.

Metoda rozproszona – wyniki wydajnościowe I [P5],[P6]

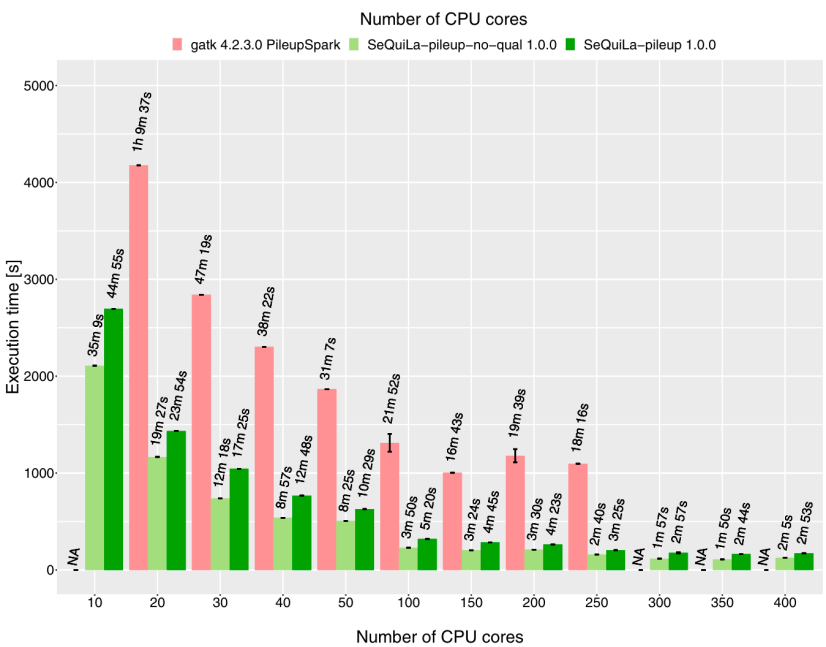
- $\sim$ 25% bardziej wydajna niż samtools[8] dla 1 wątku
- bardzo dobra skalowalność dla 1 – 10 wątków także dzięki minimalizacji wymiany danych poprzez adaptacyjne scalanie partycji wejściowych



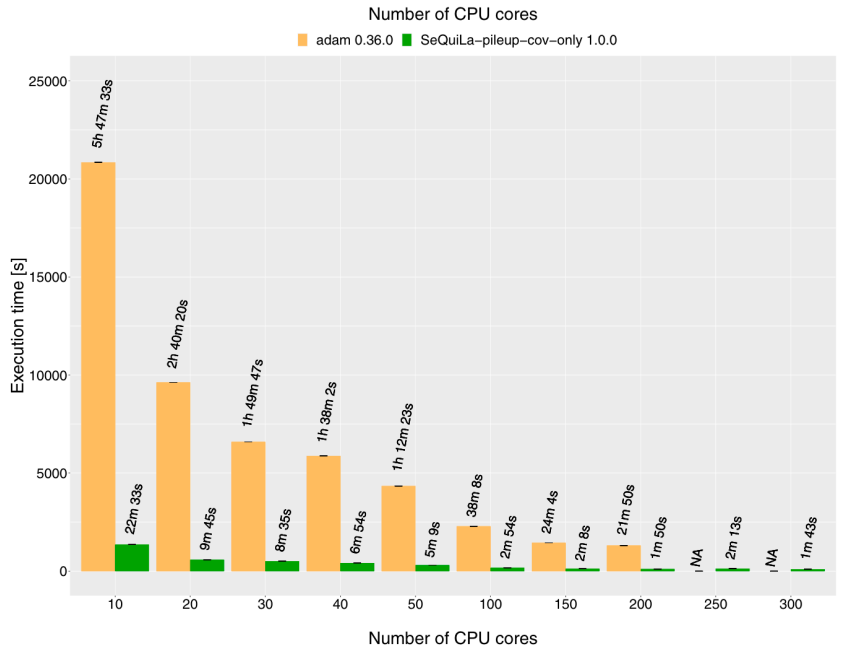
Rysunek: Porównanie wydajności ES (**A**); WGS (**B**), pojedynczy węzeł.

Metoda rozproszona – wyniki wydajnościowe II [P5],[P6]

- w przypadku obliczeń rozproszonych zdecydowanie lepsze wyniki niż GATK[9]($\sim 3-5x$) oraz ADAM[10]($\sim 11-16x$) (oba narzędzia bazujące na Apache Spark)
- $\sim 2x$ mniejsze zużycie pamięci niż GATK



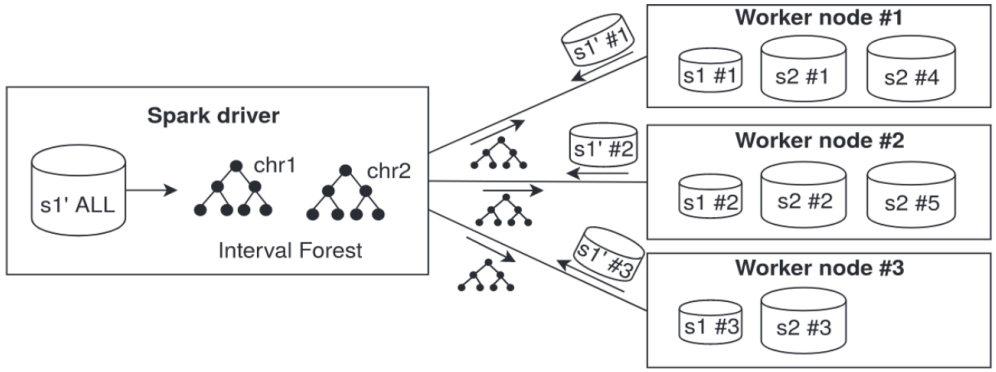
Rysunek: **Podsumowanie odczytów**, WGS, klaster Hadoop.



Rysunek: **Głębokość pokrycia**, WGS, klaster Hadoop.

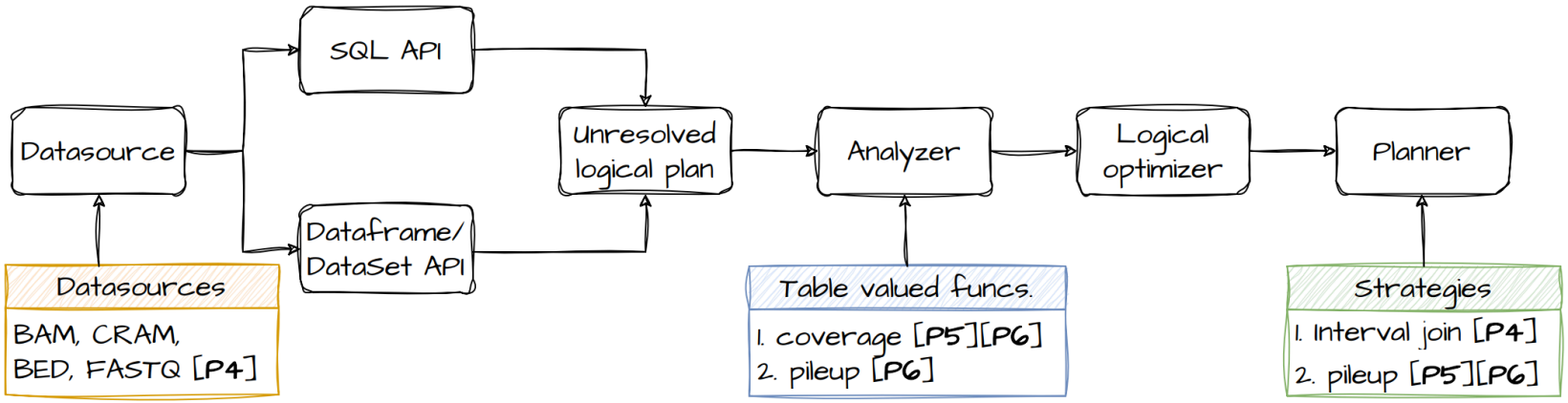
Operacje przecięć przedziałowych [P4]

```
SELECT s1.target,
       count(*)
FROM s1
JOIN s2 ON  s1.chr=s2.chr
        AND s1.end>=s2.start
        AND s1.start<=s2.end
GROUP BY s1.target
```



- **[P4]** prezentuje nowy algorytm bazujący domyślnie na Augmented Interval Tree[11]
- implementujący dwie strategie złączeń zależnych od szacowanego rozmiaru struktur do rozgłaszania (całe rekordy albo ich identyfikatory, dwuetapowe złączanie[12])
- w każdym przypadku możliwość użycia innych struktur danych, jak np. Nested Containment List[13], Augmented Interval List[14], czy Implicit Interval Tree[15]
- $\sim 2 - 5x$ szybszy niż inne rozproszone algorytmy

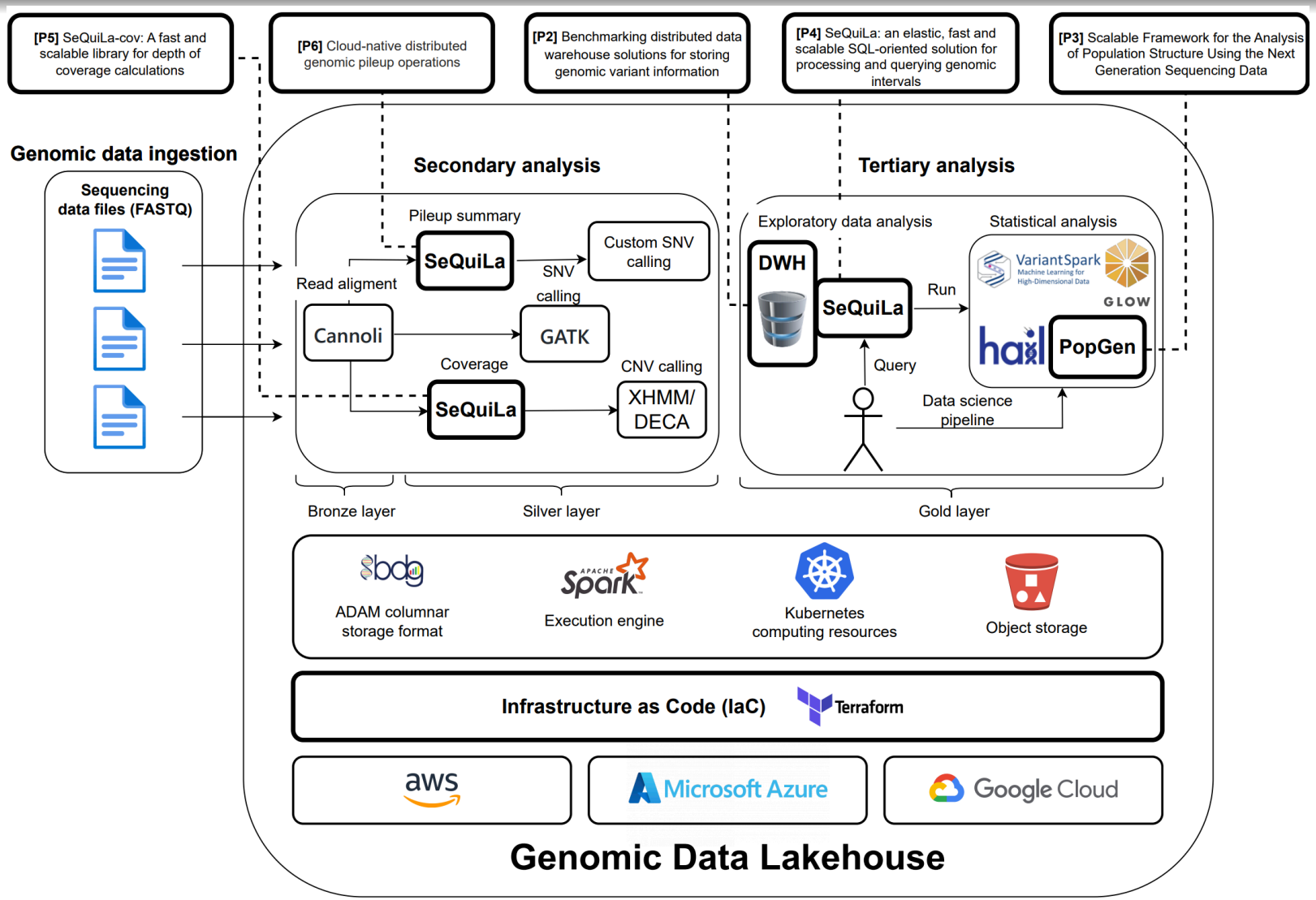
Projekt SeQuiLa [P4],[P5],[P6]



- rozwijany jako projekt na licencji Apache[1]
- rozszerzenia do optymalizatora Catalyst
- wsparcie dla bioinformatycznych formatów
- możliwość użycia z poziomu interfejsów SQL jak i Dataframe/Dataset
- wsparcie dla Scala/Java oraz Python

[1] https://biodatageeks.github.io/sequila/

Platforma typu *Genomic Data Lakehouse*



Podsumowanie

W rozprawie zostały udowodnione **wszystkie** postawione tezy badawcze, w szczególności poprzez:

1. zaproponowanie **nowej metodologii** oceny i porównania rozproszonych silników zapytań SQL do przechowywania i analizy informacji o wariantach
2. zaprojektowanie i zaimplementowanie **nowatorskich algorytmów rozproszonych**, istotnych z punktu widzenia analiz genomicznych, służących do: znajdowania **przecięć przedziałów**, obliczania **głębokości pokrycia** i uzyskiwania **podsumowania odczytów**
3. zapoczątkowanie projektu **SeQuiLa** mogącego posłużyć do realizacji platform genomicznych typu **Data Lakehouse** na bazie rozwiązania **Apache Spark**.

Dziękuję za uwagę

Odpowiedzi na pytania Recenzentów

Pytanie 1

P1:W rozdziale 2.1 autoreferatu napisano „development of novel algorithms is a lot easier than with the low-level Map-Reduce approach". Nie znalazłem jednak dla tego stwierdzenia uzasadnienia. Mogę przypuszczać dlaczego tak jest, ale chciałbym się dowiedzieć skąd zdaniem Autora wynika ta łatwość.
ODP:

- Apache Spark zapewnia wysokopoziomowe interfejsy programistyczne w językach takich jak Python, Java, Scala i R.
- MapReduce zazwyczaj korzysta z języka Java, a interfejsy programistyczne są znacznie niższego poziomu - np. konieczność tworzenia własnych implementacji podstawowych operacji jak sortowanie czy filtrowanie, agregacje, itd.
- Apache Spark zaadaptował koncepcję DataFrames i Datasets, zapewniając wyższy poziom abstrakcji do manipulacji danymi
- MapReduce wykorzystuje głównie pary klucz-wartość, które mogą być mniej intuicyjne i trudniejsze w obsłudze w przypadku złożonych operacji.

Pytanie 2

P2: Na rysunku 1.2 autoreferatu pojawia się blok „[P6] Cloud-native distributed genomic pileup operations". Co oznacza wg Autora rozprawy „cloud-native''?
ODP:
Pojęcie „cloud-native'' nie jest rzeczywiście jednoznacznie zdefiniowane w literaturze i dosyć często różnie rozumiane w zależności od kontekstu, w którym jest użyte. Wychodząc poza mocno ogólną definicję rozwiązania, które jest zaprojektowane pod kątem pełnego wykorzystania korzyści płynących z platform chmurowych, w recenzowanej pracy miałem na myśli poniższe cechy:

- możliwość wykorzystania usług zarządzanych
- możliwość uruchomienia w środowiskach kontenerowych (np. Kubernetes)
- wsparcie dla rozdzielenia warstw obliczeń od warstwy przechowywania danych, tj. możliwość niezależnego skalowania obu elementów
- sposób instalacji wspierający podejście „infrastruktura jako kod"

Pytanie 3

P3: W odniesieniu do publikacji [P2] i użytych w prowadzonych badaniach formatów danych ORC i Parquet, czy w świetle wyników badań byłoby rozsądne zaproponowanie własnego formatu przechowywania danych dla pewnych grup danych pod kątem analiz genomicznych wykonywanych na platformach klasy Big Data jak Apache Spark czy Hadoop?
ODP:

- Generyczne kolumnowe formaty danych jak ORC są bardzo dobrze zoptymalizowane pod kątem przetwarzania dużych wolumenów danych w środowiskach rozproszonych.
- Podobnie ma się sytuacja z magazynami danych, na których te zbiory w omawianych formatach są przechowywane - są zoptymalizowane pod sekwencyjny odczyt dużych bloków danych, dostęp losowy jest zdecydowanie mniej wydajny.
- W przypadku danych zagregowanych oraz mocno selektywnych zapytań wykorzystujących filtrowanie w oparciu o wiele atrybutów zdecydowanie wartym rozważenia jest wykorzystywanie istniejących rozwiązań wspierających wydajne mechanizmy indeksowania.
- Najlepszym w takim przypadku wydaje się podejście hybrydowe w postaci federacyjnych silników zapytań, które sięgają do baz danych zoptymalizowanych pod różne ich charakterystyki.

Pytanie 4

P4: Moją ciekawość zawsze budzi stwierdzenie „głębokość pokrycia (depth of coverage)'' i zawsze pojawia się u mnie pytanie „pokrycia, ale czego?''
ODP:
Pojęcie „głębokości pokrycia'' jest rzeczywiście często używane w odniesieniu do liczby odczytów z procesu sekwencjonowania pokrywających daną pozycję bez specyfikowania, czy mamy do czynienia z genomem (analiza danych z DNA-Seq), czy też np. z transkryptomem (RNA-Seq). W przypadku recenzowanej pracy odnosiło się to zawsze do genomu.

Pytanie 5

P5: W odniesieniu do publikacji [P6], w autoreferacie stwierdzono „implementation of Apache Spark Catalyst custom execution strategy that handles depth of coverage calculations for both DataFrame and SQL APIs''. Na czym dokładnie polegała ta strategia?

ODP:

Została przedstawiona dokładniej na slajdzie pt. „Metoda rozproszona [P5], [P6]"

1. wyznaczenie wirtualnych i scalonych partycji dla plików BAM/CRAM
2. wyznaczenie dla każdej wirtualnej partycji struktury zawierającej zdarzenia początku i końca odczytów
3. wyznaczanie struktur dla alternatywnych alleli na podstawie pól MD i CIGAR
4. generowanie wyników (głębokość pokrycia, podsumowanie odczytów) w postaci tabeli

Pytanie 6

P6: Odnośnie osiągnięć O22 i O23 przedstawionych w sekcji 3.8 autoreferatu, w jakim konkursie je otrzymano? Czy był to wewnętrzny konkurs Politechniki Warszawskiej?
ODP:
ODP: Tak, to była I edycja konkursu Best Paper Politechniki Warszawskiej.

Pytanie 1

P1: Układ pracy a czytelność.
ODP:
Zgadzam się z uwagami dotyczącymi możliwości dokonania alternatywnego podziału układu pracy oraz umieszczenia wymienionych rozdziałów (tj. opisu dorobku, kopii prac oraz oświadczeń współautorów) jako załączników.

Pytanie 2

P2: Wpływ prezentowanych metod i algorytmów na wyniki badań bioinformatycznych oraz ich stosowalność.
ODP:
Zgadzam się, iż omawiane rozproszone metody i algorytmy nie reprezentują nowych sposobów analizy danych z wysokoprzepustowego sekwencjonowania **per se**.

- Taka też była główna idea prowadzonych przeze mnie badań - opracować **nowe, rozproszone i wydajne algorytmy** dla istniejących, podstawowych operacji genomicznych, które mogą być używane w **środowiskach chmurowych**.
- Uwypuklona w pracy skalowalność i wydajność przekładająca się bezpośrednio na możliwość analizy danych w większej skali, większych populacji, dla projektów takich jak gnomAD, „1000 polskich genomów".
- Prezentowane rozwiązania,chociaż możliwe do wykorzystania w dowolnej skali badań, są przede wszystkim przeznaczone do wsparcia badań w **dużych projektach genomicznych**.

Pytanie 3

P3: Problem upowszechniania prezentowanych rozwiązań w społeczności rzeczywistych użytkowników (bioinformatyków) pracujących w centrach sekwencjonowania.
ODP:
Problem ten jest zarówno bardzo istotny, jak i wieloaspektowy.

- Rozwiązania realizujące obliczenia rozproszone wymagają z definicji dużo bardziej złożonej infrastruktury informatycznej aniżeli rozwiązania jednowęzłowe.
- Jednym z aspektów poruszanych w recenzowanej pracy, jak i w publikacji [P6], było **podejście infrastruktura jako kod**, które zwłaszcza w przypadku środowisk chmur prywatnych i publicznych może w znaczący sposób obniżyć barierę wejścia i tym samym przyczynić się do upowszechniania bardziej skomplikowanych narzędzi.
- Inny aspekt, który jest poruszany w pracach P4–P6, to **standaryzacja interfejsów programistycznych w postaci** np. języka SQL i abstrakcji ramki danych (DataFrame'a), tak aby umożliwić łatwą analizę danych genetycznych przez osoby mniej techniczne

Odpowiedzi na pytania Recenzentów

Bibliografia I

[1] Matti Niemenmaa, Aleksi Kallio, André Schumacher, Petri Klemelä, Eija Korpelainen, and Keijo Heljanko. Hadoop-BAM: directly manipulating next generation sequencing data in the cloud. *Bioinformatics*, 28(6):876–877, March 2012. ISSN 1460-2059. doi: 10.1093/bioinformatics/bts054.

[2] André Schumacher, Luca Pireddu, Matti Niemenmaa, Aleksi Kallio, Eija Korpelainen, Gianluigi Zanetti, and Keijo Heljanko. SeqPig: simple and scalable scripting for large sequencing data sets in Hadoop. *Bioinformatics*, 30(1):119–120, January 2014. ISSN 1460-2059. doi: 10.1093/bioinformatics/btt601.

[3] Raghunath Othayoth Nambiar and Meikel Poess. The making of TPC-DS. In *Proceedings of the 32nd international conference on very large data bases*, VLDB '06, pages 1049–1058. VLDB Endowment, 2006. Place: Seoul, Korea Number of pages: 10.

[4] Xiaoming Liu, Chunlei Wu, Chang Li, and Eric Boerwinkle. dbNSFP v3.0: A One-Stop Database of Functional Predictions and Annotations for Human Nonsynonymous and Splice-Site SNVs. *Human Mutation*, 37(3):235–241, March 2016. ISSN 10597794. doi: 10.1002/humu.22932. URL `https://onlinelibrary.wiley.com/doi/10.1002/humu.22932`.

Bibliografia II

[5] Exome Aggregation Consortium, Monkol Lek, Konrad J. Karczewski, Eric V. Minikel, Kaitlin E. Samocha, Eric Banks, Timothy Fennell, Anne H. O'Donnell-Luria, James S. Ware, Andrew J. Hill, Beryl B. Cummings, Taru Tukiainen, Daniel P. Birnbaum, Jack A. Kosmicki, Laramie E. Duncan, Karol Estrada, Fengmei Zhao, James Zou, Emma Pierce-Hoffman, Joanne Berghout, David N. Cooper, Nicole Deflaux, Mark DePristo, Ron Do, Jason Flannick, Menachem Fromer, Laura Gauthier, Jackie Goldstein, Namrata Gupta, Daniel Howrigan, Adam Kiezun, Mitja I. Kurki, Ami Levy Moonshine, Pradeep Natarajan, Lorena Orozco, Gina M. Peloso, Ryan Poplin, Manuel A. Rivas, Valentin Ruano-Rubio, Samuel A. Rose, Douglas M. Ruderfer, Khalid Shakir, Peter D. Stenson, Christine Stevens, Brett P. Thomas, Grace Tiao, Maria T. Tusie-Luna, Ben Weisburd, Hong-Hee Won, Dongmei Yu, David M. Altshuler, Diego Ardissino, Michael Boehnke, John Danesh, Stacey Donnelly, Roberto Elosua, Jose C. Florez, Stacey B. Gabriel, Gad Getz, Stephen J. Glatt, Christina M. Hultman, Sekar Kathiresan, Markku Laakso, Steven McCarroll, Mark I. McCarthy, Dermot McGovern, Ruth McPherson, Benjamin M. Neale, Aarno Palotie, Shaun M. Purcell, Danish Saleheen, Jeremiah M. Scharf, Pamela Sklar, Patrick F. Sullivan, Jaakko Tuomilehto, Ming T. Tsuang, Hugh C. Watkins, James G. Wilson, Mark J. Daly, and Daniel G. MacArthur. Analysis of protein-coding genetic variation in 60,706 humans. *Nature*, 536(7616):285–291, August 2016. ISSN 0028-0836,

Bibliografia III

1476-4687. doi: 10.1038/nature19057. URL http://www.nature.com/articles/nature19057.

[6] Joelle Mbatchou, Leland Barnard, Joshua Backman, Anthony Marcketta, Jack A. Kosmicki, Andrey Ziyatdinov, Christian Benner, Colm O'Dushlaine, Mathew Barber, Boris Boutkov, Lukas Habegger, Manuel Ferreira, Aris Baras, Jeffrey Reid, Goncalo Abecasis, Evan Maxwell, and Jonathan Marchini. Computationally efficient whole-genome regression for quantitative and binary traits. *Nature Genetics*, 53(7):1097–1103, July 2021. ISSN 1061-4036, 1546-1718. doi: 10.1038/s41588-021-00870-7. URL https://www.nature.com/articles/s41588-021-00870-7.

[7] Brent S. Pedersen and Aaron R. Quinlan. Mosdepth: quick coverage calculation for genomes and exomes. *Bioinformatics*, 34(5):867–868, March 2018. ISSN 1367-4803. doi: 10.1093/BIOINFORMATICS/BTX699. URL https://academic.oup.com/bioinformatics/article/34/5/867/4583630. Publisher: Oxford Academic.

Bibliografia IV

[8] Petr Danecek, James K Bonfield, Jennifer Liddle, John Marshall, Valeriu Ohan, Martin O Pollard, Andrew Whitwham, Thomas Keane, Shane A McCarthy, Robert M Davies, and Heng Li. Twelve years of SAMtools and BCFtools. *GigaScience*, 10(2), January 2021. ISSN 2047-217X. doi: 10.1093/gigascience/giab008. URL `https://academic.oup.com/gigascience/article/doi/10.1093/gigascience/giab008/6137722`.

[9] Aaron McKenna, Matthew Hanna, Eric Banks, Andrey Sivachenko, Kristian Cibulskis, Andrew Kernytsky, Kiran Garimella, David Altshuler, Stacey Gabriel, Mark Daly, and Mark A. DePristo. The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Research*, 20(9):1297–1303, September 2010. ISSN 1088-9051. doi: 10.1101/gr.107524.110.

[10] Matt Massie, Frank Nothaft, Christopher Hartl, Christos Kozanitis, André Schumacher, Anthony D Joseph, and David A Patterson. Adam: Genomics formats and processing patterns for cloud scale computing. *University of California, Berkeley Technical Report, No. UCB/EECS-2013*, 207:2013, 2013.

[11] Thomas H. Cormen, Charles Eric Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to algorithms*. The MIT Press, Cambridge, Massachusett, fourth edition edition, 2022. ISBN 978-0-262-36750-9. OCLC: 1305060400.

Bibliografia V

[12] Christos Kozanitis and David A. Patterson. GenAp: A distributed SQL interface for genomic data. *BMC Bioinformatics*, 17(1):1–8, February 2016. ISSN 14712105. doi: 10.1186/S12859-016-0904-1/TABLES/2. URL https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-016-0904-1. Publisher: BioMed Central Ltd.

[13] Alexander V. Alekseyenko and Christopher J. Lee. Nested Containment List (NCList): A new algorithm for accelerating interval query of genome alignment and interval databases. *Bioinformatics*, 23(11):1386–1393, June 2007. ISSN 13674803. doi: 10.1093/BIOINFORMATICS/BTL647.

[14] Jianglin Feng, Aakrosh Ratan, and Nathan C. Sheffield. Augmented Interval List: A novel data structure for efficient genomic interval search. *Bioinformatics*, 35 (23):4907–4911, December 2019. ISSN 14602059. doi: 10.1093/BIOINFORMATICS/BTZ407. Publisher: Oxford University Press.

[15] Heng Li and Jiazhen Rong. Bedtk: finding interval overlap with implicit interval tree. *Bioinformatics*, 37(9):1315–1316, June 2021. ISSN 1367-4803. doi: 10.1093/BIOINFORMATICS/BTAA827. URL https://academic.oup.com/bioinformatics/article/37/9/1315/5910546. Publisher: Oxford Academic.