Pipeline Filogenetyczny 2.0

Contents

1. Wstęp

Moją pracę oparłam na artykule badającym relacje pomiędzy rodzajami Brettanomyces, Debaryomyces, Dekkera oraz Kluyveromyces [1]. Wybrałam z niego 20 gatunków drożdży, starając się wybierać je w miarę proporcjonalnie do rozmiaru kladu, dbając jednocześnie o dostępność genomów w bazie NCBI. W powyrzszym artyluke, autorzy inferowali drzewo filegenetyczne, bazując na 18S rRNA dla 116 genomów drożdży, z czego 28 z nich została zsekwencjonowana w ramach badania; pozostałe sekwencje pochodziły z baz GenBank oraz EMBL. Same drzewa zostały stworzone w oparciu o metodę NJ w narzędziu PHYLIP v3.5 po uliniowieniu sekwencji programem PILEUP i ręcznym dopasowaniu. Poziomy ufności zostały określone metodą bootstrap z 500 replikami.

Drzewo do porównania z literaturą napisałam ręcznie, pomijając wartości wsparcia. Do porównania użyłam również taksonomii NCBI [2] oraz Time Tree [3].

2. Metody

1. W moim pipeline'ie bazowałam na proteomach z bazy NCBI [4], pobranych przy użyciu narzędzia ncbi datasets [5].
2. Tak pobrane proteomy poklastrowałam programem MMSeqs2 [6], korzystając z klastrowania hierarchicznego, ponieważ dane nie były zbyt duże. Zmniejszyłam tu wartości parametrów `--min-seq-id 0.5` oraz `-c 0.7` w celu zwiększenia ilości klastrów ortologicznych.
3. Z obliczonych klastrów generowałam klastry ortologiczne. W tym celu skorzystałam z własnego skryptu w języku python (`make_orth.py`), działającego wielowątkowo z pomocą biblioteki tqdm [7]. Pomijałam tu klastry, które nie zawierały przynajmniej jednego reprezentanta dla każdego taksonu. W przypadku klastrów ortologicznych, zachłannie wybierałam sekwencje o długości najbliższej średniej długości sekwencji już wybranych, uporządkowanych w kolejności malejącej względem długości. W przypadku kilku sekwencji dla pierwszego taksonu w tej kolejności, wybierałam je losowo z rozkładu jednostajnego. Na tym etapie również zmieniałam etykiety sekwencji na nazwy taksonów.
4. Tak wygenerowane klastry uliniowiałam programem [8], korzystając z domyślnych opcji. Użyłam dodatkowo programu GNU parallel [9] w celu przyspieszenia obliczeń, wykorzystując

wszystkie rdzenie procesora. Programu tego używałam również na wielu innych etapach pipeline'u.

5. Na bazie uliniowionych klastrów, liczyłam drzewa metodą ML z programu IQ-TREE 3 [10]. W celach optymalizacyjnych, wybrałam najczęściej wybierane modele ewolucyjne na podstawie histogramu bazującego na najlepszych modelach dla pierwszych 50 drzew. W rezultacie, otrzymałam modele `LG+I+G4, Q.PFAM+I+G4, Q.PFAM+F+I+G4, Q.YEAST+I+G4, Q.YEAST+F+I+G4`. Zredukowało to ilość sprawdzanych modeli z ~ 1000 do ~ 70. W ramach punktu V.a, liczę tu również 1000 drzew bootstrap, korzystając z metody Ultrafast Bootstrap, gdyż metoda Bootstrap nie była obliczalna w sensownym czasie na moim komputerze.
6. Drzewa zbudowane na klastrach ortologicznych filtrowałam na podstawie minimalnego wsparcia UFBoot/SH-aLRT na poziomie >30%, korzystając z IQ-TREE 3 oraz własnego skryptu w pythonie, sprawdzającego powyższy warunek (`filter_supported.py`). Przez filtr przeszło 202/241 drzew.
7. Na podstawie drzew ML dla klastrów ortologicznych oraz przefiltrowanych drzew ML dla klastrów ortologicznych, liczyłam drzewa konsensusowe metodą extended majority (domyślną).
8. Superdrzewa liczyłam na podstawie drzew ML dla klastrów ortologicznych. W tym celu, użyłam programu Clann [11] z kryterium dfit [12], heurystyką wyszukiwania SPR, 20 krokami oraz 5 repetycjami.
9. Do porównania otrzymanych drzew użyłam biblioteki TreeDist [13], [14], [15] w R, korzystając z unormowanej odległości Robinsona-Fouldsa oraz unormowanej odległości Jaccarda-Robinsona-Fouldsa. Ze względu na brak wystarczających informacji o taksonach *Rhodotorula glutinis i Sporobolomyces roseus* w bazie TimeTree, musiałam je usunąć z drzew wynikowych. W tym celu użyłam pakietu biopython [16], dokładnie metody `Bio.Phylo.Newick.Tree.prune(<missing_taxon>)` oraz usunęłam długości krawędzi z obydwu drzew poleceniem `sed`, gdyż zostały one wyzerowane po operacji `prune`, co dawało brzydki wykres porównania.

Do samej organizacji zadań użyłam programu GNU Make, który wymaga ustawienia zmiennej środowiskowej `NCBI_API_KEY`. Do uruchomienia pipeline'u powinno wystarczyć wykonanie polecenia `make`. Przy urównoleglaniu pipeline'u zdecydowałam się na uliniowianie warstwowe, tj. w obrębie jednego zadania, z użyciem GNU Parallel, gdyż wykonujemy serię bardzo podobnych poleceń, które czasami wymagają wszystkich wyników z poprzedniego zadania (np. liczenie drzew konsensusowych wymaga policzenia wszystkich drzew ML dla klastrów ortologicznych). Skrypt ten nie jest zatem przystosowany do uruchamiania go z parametrem `-j n` dla $n > 1$.

3. Wyniki

3.1. Środowisko

Pipeline został uruchomiony na moim komputerze z procesorem AMD Ryzen 7 5700U (16) @ 4.3GHz, 30.69 GiB dostępnej pamięci RAM DDR4, na systemie NixOS 26.05 z kernelem 6.18.0-zen1.

Długość całych obliczeń trwała ok. 2.5h, z czego liczenie drzew ML zajęło ok. 1.5h.

3.2. Rozkład klastrów

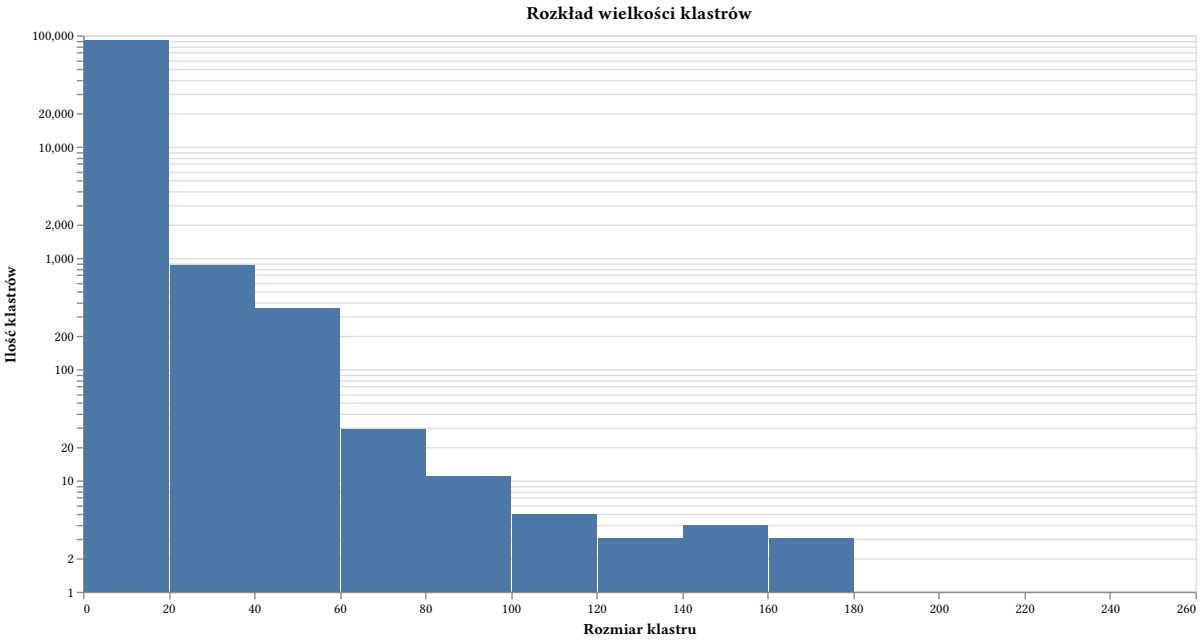

Figure 1: Rozkład rozmiaru wszystkich klastrów w skali logarytmicznej. Pierwszy kubełek odpowiada klastrom, które na pewno zostały odrzucone. Najmniejszy klaster miał wielkość 2, a największy — 248.

3.3. Porównania

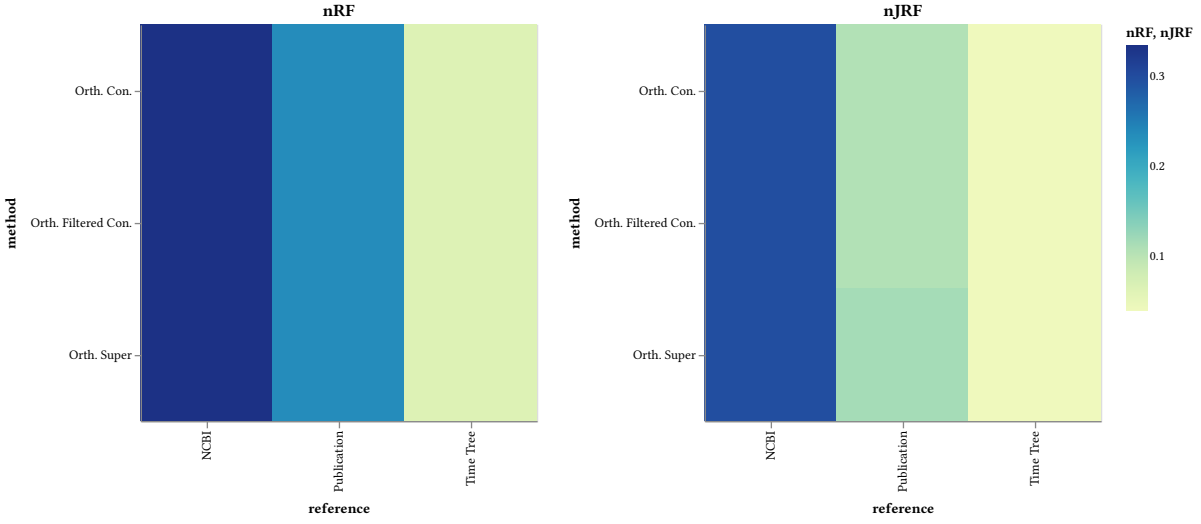

Figure 2: Porównania różnych metod inferencji z różnymi referencjami

method	reference	nRF	nJRF
Orth. Con.	NCBI	0.333333	0.296296
Orth. Con.	Publication	0.235294	0.106209
Orth. Con.	Time Tree	0.066667	0.040000
Orth. Filtered Con.	NCBI	0.333333	0.296296
Orth. Filtered Con.	Publication	0.235294	0.106209
Orth. Filtered Con.	Time Tree	0.066667	0.040000
Orth. Super	NCBI	0.333333	0.296296
Orth. Super	Publication	0.235294	0.116013
Orth. Super	Time Tree	0.066667	0.040000

Table 1: Tabela porównań uzyskanych drzew z drzewami referencyjnymi.

3.3.1. Wizualizacje TreeDist

3.3.1.1. Drzewo konsensusowe, klastry ortologiczne i filtrowane

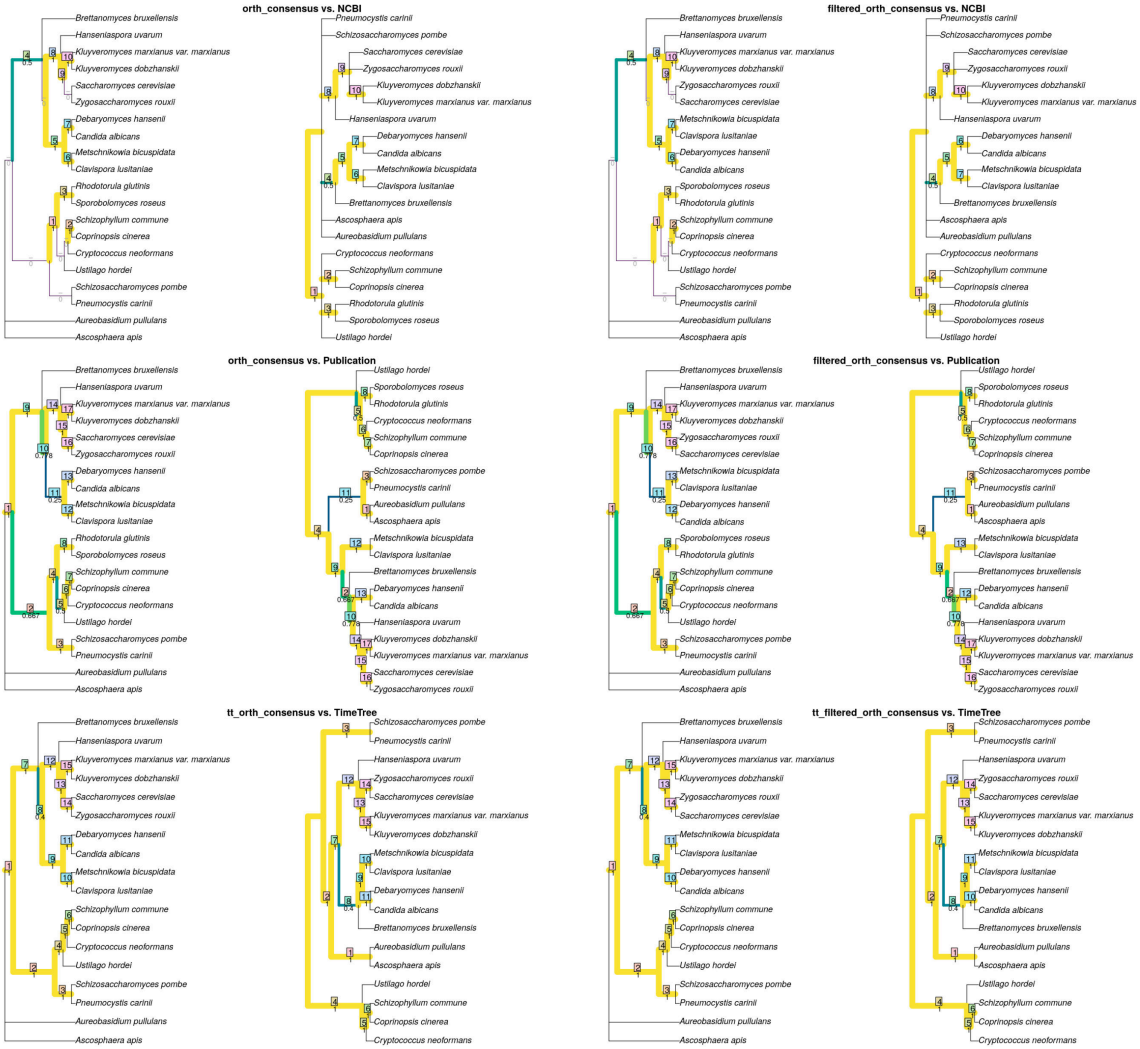

Figure 3: Porównania drzew konsensusowych z klastrów ortologicznych (lewo) i drzew konsensusowych z przefiltrowanych drzew na klastrach ortologicznych (prawo) z drzewami referencyjnymi, używając miary nJRF.

3.3.1.2. Super drzewo

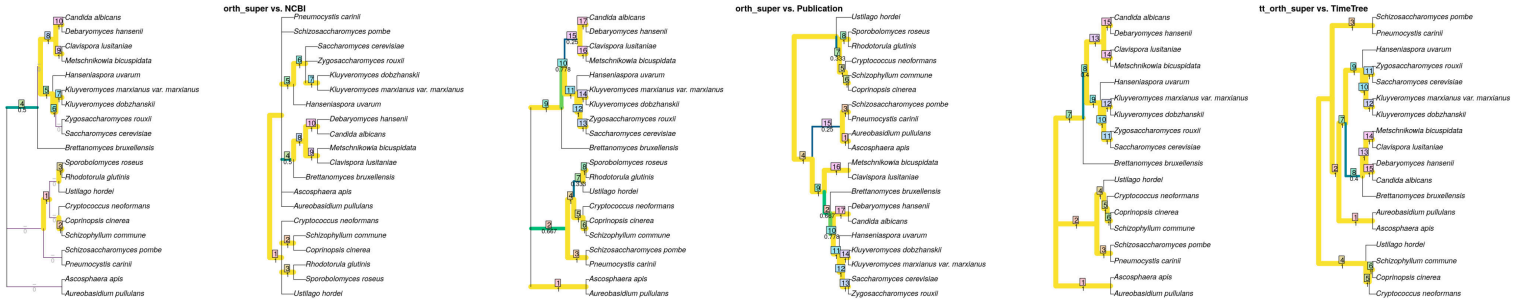

Figure 4: Porównanie super drzewa z drzew ML na klastrach ortologicznych z drzewami referencyjnymi, używając miary nJRF.

4. Wnioski

4.1. Podsumowanie

Wyniki przejawiają bardzo wysokie podobieństwo do drzewa z bazy TimeTree, które wydaje się być tu preferowane ze względu na ograniczony zbiór danych w oryginalnej publikacji oraz wątpliwą jakość taksonomii NCBI (w tym przypadku).

4.2. Uwagi

4.2.1. Ograniczenia modeli

Warte może być zezwolenie na wybór modelu z większej puli modeli podczas liczenia drzew ML z IQ-TREE. Tak silne ograniczenie wyboru modelu jak tutaj może prowadzić do mniej reprezentatywnych drzew ML, wpływając w znacznym stopniu na jakość wyników końcowych.

Samo ograniczenie zostało wybrane tu w celu optymalizacji czasu działania pipeline'u, gdyż Model Finder zajmował znaczną część czasu liczenia drzew ML.

4.2.2. Bootstrap

Ponownie ze względów optymalizacyjnych, został tu użyty Ultrafast Bootstrap zamiast nieparametrycznego Bootstrapu. Również mogło wpłynąć to na jakość wyników, ponieważ UFBoot używa zaledwie przybliżenia MLE.

Jednakże, użycie zwykłego bootstrapu prowadziłoby tu do ok. 100-1000x wydłużenia czasu działania pipeline'u (w zależności od liczby powtórzeń), co nie było tu praktyczne.

4.2.3. Ponowne klastrowanie

Jak widać na rozkładzie wielkości klastrów Figure 1, powalająca większość klastrów zostala odrzucona przy generowaniu klastrów ortologicznych ze względy na brak reprezentantów dla każdego taksonu. Jednym z potencjalnych rozwiązań byłoby tu ponowne poklastrowanie takich klastrów, używając luźniejszych kryteriów klastrowania, np. zmniejszając wartości parametrów `-c` oraz `--min-seq-id`. Taka operacja mogłaby zostać powtórzona kilkukrotnie.

Sama korzyść płynąca z dodania nowych, mniej konserwatywnych klastrów pozostaje do zbadania.

Bibliografia

[1] J. CAI, I. N. ROBERTS, and M. D. COLLINS, "Phylogenetic Relationships among Members of the Ascomycetous Yeast Genera Brettanomyces, Debaryomyces, Dekkera, and Kluyveromyces Deduced by Small-Subunit rRNA Gene Sequences," *International Journal of Systematic and Evolutionary Microbiology*, vol. 46, no. 2, pp. 542–549, 1996, doi: https://doi.org/10.1099/00207713-46-2-542.

[2] C. L. Schoch *et al.*, "NCBI Taxonomy: a comprehensive update on curation, resources and tools," *Database*, vol. 2020, p. baaa62, 2020, doi: 10.1093/database/baaa062.

[3] S. Kumar *et al.*, "TimeTree 5: An Expanded Resource for Species Divergence Times," *Molecular Biology and Evolution*, vol. 39, no. 8, p. msac174, 2022, doi: 10.1093/molbev/msac174.

[4] E. W. Sayers, M. Cavanaugh, K. Clark, J. Ostell, K. D. Pruitt, and I. Karsch-Mizrachi, "GenBank," *Nucleic Acids Research*, vol. 47, no. D1, pp. D94–D99, 2018, doi: 10.1093/nar/gky989.

[5] N. A. O'Leary *et al.*, "Exploring and retrieving sequence and metadata for species across the tree of life with NCBI Datasets," *Scientific Data*, vol. 11, no. 1, p. 732, Jul. 2024, doi: 10.1038/s41597-024-03571-y.

[6] M. Steinegger and J. Söding, "MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets," *Nature Biotechnology*, vol. 35, no. 11, pp. 1026–1028, Nov. 2017, doi: 10.1038/nbt.3988.

[7] C. da Costa-Luis, "tqdm: A Fast, Extensible Progress Meter for Python and CLI," *Journal of Open Source Software*, vol. 4, p. 1277, 2019, doi: 10.21105/joss.01277.

[8] R. C. Edgar, "Muscle5: High-accuracy alignment ensembles enable unbiased assessments of sequence homology and phylogeny," *Nature Communications*, vol. 13, no. 1, p. 6968, Nov. 2022, doi: 10.1038/s41467-022-34630-w.

[9] O. Tange, "GNU Parallel 20251222 ('Bondi')." [Online]. Available: https://doi.org/10.5281/zenodo.18039569

[10] H. R. H. B. A. J. R. E. S. C. B. N. D. M. N. G. M. W. H. G. H. R. L. B. Q. M. Thomas K.F. Wong Nhan Ly-Trong, " IQ-TREE 3: Phylogenomic Inference Software using Complex Evolutionary Models ," 2025, [Online]. Available: https://doi.org/10.32942/X2P62N

[11] C. J. Creevey and J. O. McInerney, "Clann: investigating phylogenetic information through supertree analyses," *Bioinformatics*, vol. 21, no. 3, pp. 390–392, 2004, doi: 10.1093/bioinformatics/bti020.

[12] C. J. Creevey *et al.*, "Does a tree–like phylogeny only exist at the tips in the prokaryotes? ," *Proceedings of the Royal Society B: Biological Sciences*, vol. 271, no. 1557, pp. 2551–2558, 2004, doi: 10.1098/rspb.2004.2864.

[13] M. R. Smith, "Information theoretic Generalized Robinson-Foulds metrics for comparing phylogenetic trees," *Bioinformatics*, vol. 36, no. 20, pp. 5007–5013, 2020, doi: 10.1093/bioinformatics/btaa614.

[14] M. R. Smith, "Robust analysis of phylogenetic tree space," *Systematic Biology*, vol. 71, no. 5, pp. 1255–1270, 2022, doi: 10.1093/sysbio/syab100.

[15] M. R. Smith, "TreeDist: Distances between Phylogenetic Trees. R package version 2.11.1." 2020. doi: 10.5281/zenodo.3528124.

[16] P. J. A. Cock *et al.*, "Biopython: freely available Python tools for computational molecular biology and bioinformatics," *Bioinformatics*, vol. 25, no. 11, pp. 1422–1423, 2009, doi: 10.1093/bioinformatics/btp163.