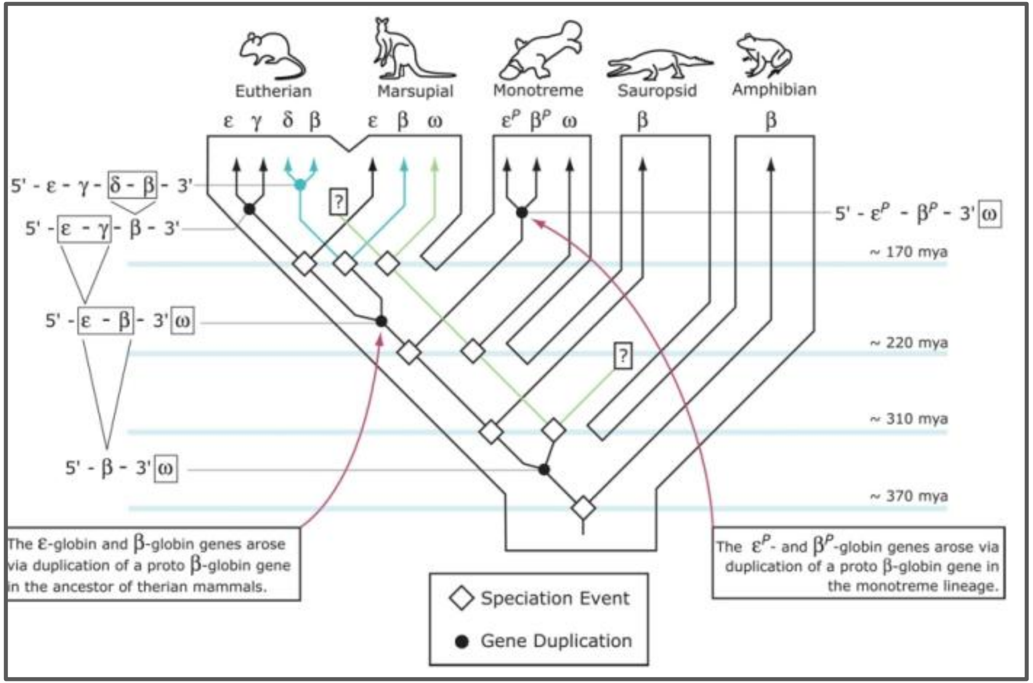
An introduction to gene families

Cedric Chauve
Department of Mathematics
Simon Fraser University

What is a (homologous) gene family?

A gene family **within a group S of species** is a **group of genes** that have evolved from a unique ancestral gene from the **Last Common Ancestor of S**, through **speciation**, **gene duplication**, **gene loss**, **gene transfer.**



The ε-globin and β-globin genes arose via duplication of a proto β-globin gene in the ancestor of therian mammals.

The ε^P- and β^P-globin genes arose via duplication of a proto β-globin gene in the monotreme lineage.

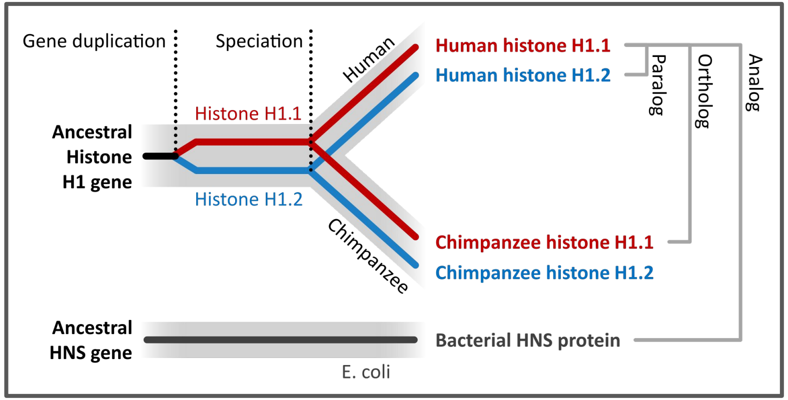
Remark. This definition occults possible issues with HGT from an unsampled outgroup species.

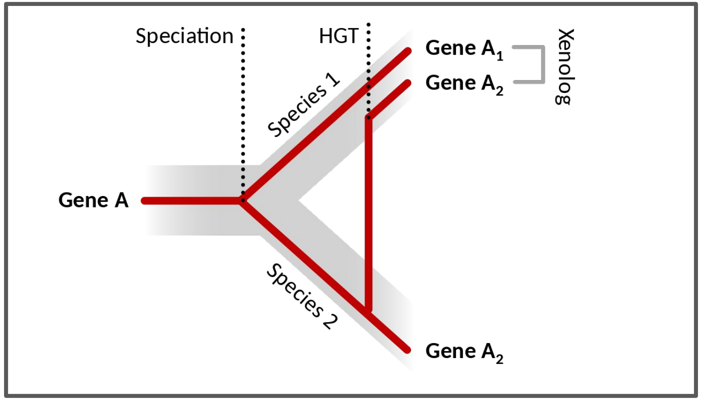
Genes within a homologous family are assumed to have maintained significant **sequence similarity, but also other genomic features,** and often to have **similar or related biological functions.**

Orthologs, paralogs, xenologs

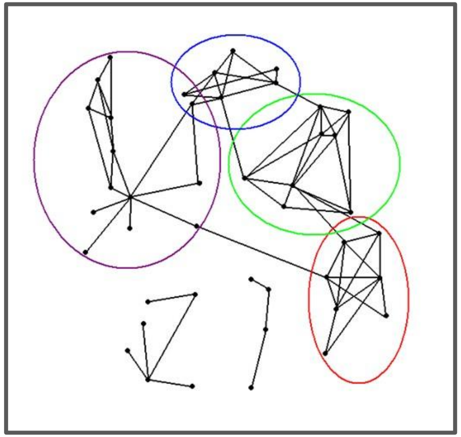
A gene family is naturally associated to a **gene tree G** that describes their evolution.

Orthologs: pair of genes with no HGT on their path in **G** and whose **LCA** is a **speciation**.
Paralogs: pair of genes with no HGT on their path in **G** and whose **LCA** is a **duplication**.
Within the same species: **in-paralogs**, in different species: **out-paralogs.**
Xenologs: pair of genes with an HGT on their path in **G.**
[Fitch 2000; Darby, 2017]





Computing gene families: a clustering problem



The problem of finding gene families from a set of genomes is often seen as a **graph clustering problem.**

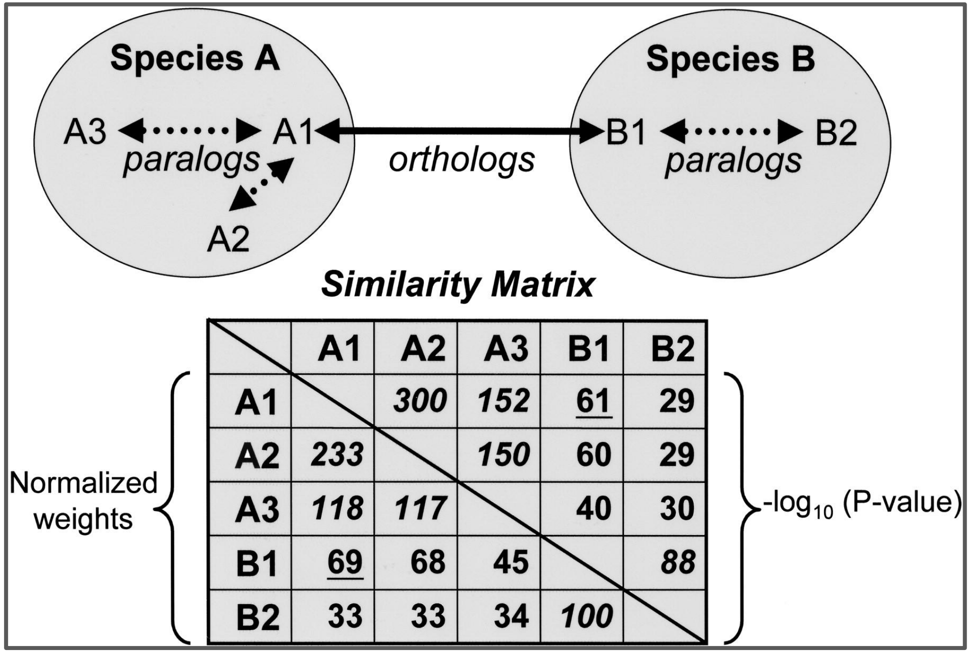
We are given a set of elements (genes of several species), forming the vertices of a graph.
Edges represent some sort of **similarity**.

We want to find groups of genes that are similar between them and sufficiently dissimilar from the other groups (or dense clusters that are weakly linked to other clusters).

Questions.

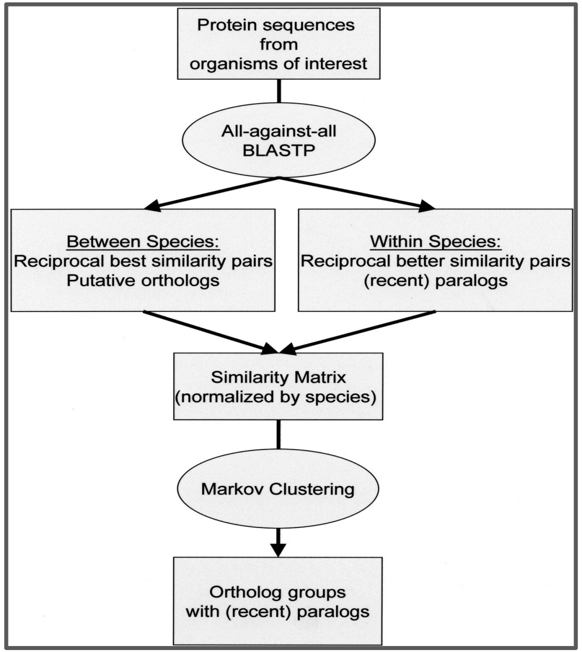
Which similarity signal should we consider?
How should we cluster?
Is-there other non-graph representable signal we should consider?

A sequence-based method: OrthoMCL (Li et al, 2003)



Similarity Matrix

	A1	**A2**	**A3**	**B1**	**B2**
A1		***300***	***152***	**61**	**29**
A2	***233***		***150***	**60**	**29**
A3	***118***	***117***		**40**	**30**
B1	**69**	**68**	**45**		***88***
B2	**33**	**33**	**34**	***100***	



Protein sequences from organisms of interest

All-against-all BLASTP

Between Species: Reciprocal best similarity pairs Putative orthologs

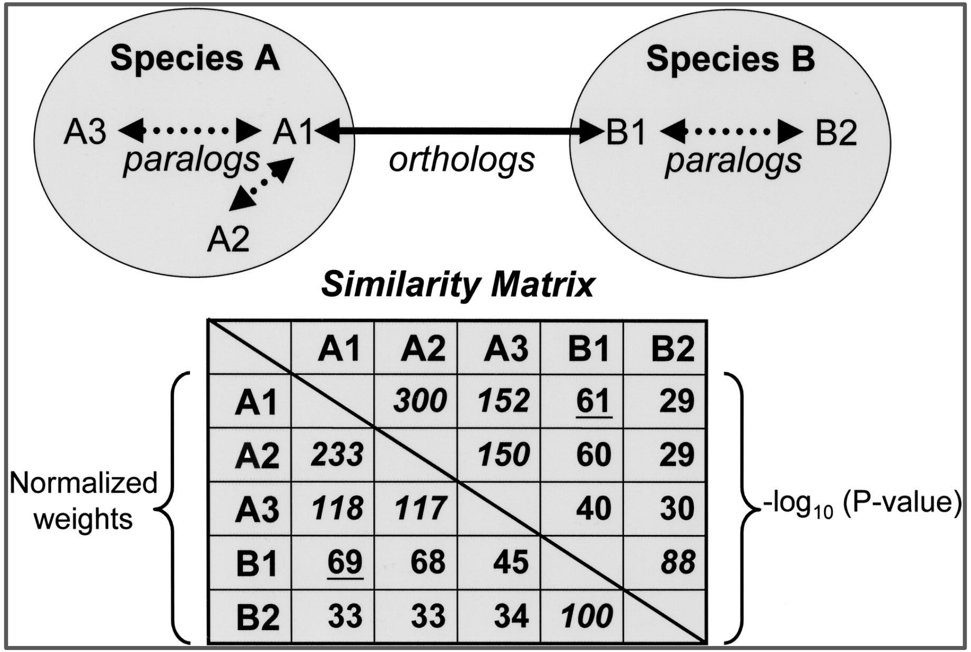
Within Species: Reciprocal better similarity pairs (recent) paralogs

Similarity Matrix (normalized by species)

Markov Clustering

Ortholog groups with (recent) paralogs

OrthoMCL: graph construction



Similarity Matrix

	A1	A2	A3	B1	B2
A1		*300*	*152*	61	29
A2	*233*		*150*	60	29
A3	*118*	*117*		40	30
B1	69	68	45		*88*
B2	33	33	34	*100*	

Orthologs: best-reciprocal hits: A1 best hit of B1 in species A and B1 best hit of A1 in species B.

In-paralogs: sequences within the same genome that are (reciprocally) more similar to each other than either is to any sequence from another genome.

Similarity score: P-value of the pairwise alignment.

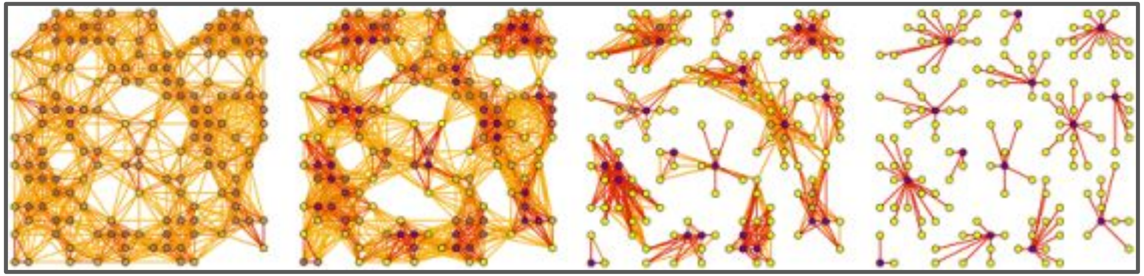
Normalization: to account for the high similarity of in-paralogs.

OrthoMCL: (Markov) clustering

Principle: A random walk is initially more likely to stay within a cluster than to leave a cluster. So often visited edges are within clusters and rarely visited edges are between clusters. But this effect tames with the length of the walk.

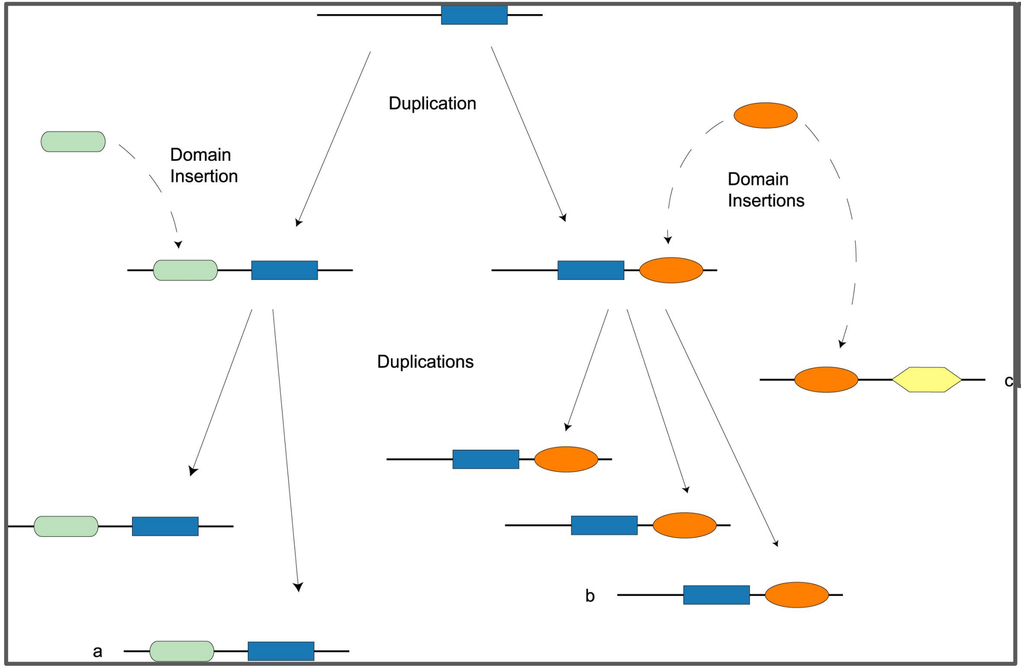
Input is an undirected graph, power parameter *e*, and inflation parameter *r*.
1. Create the associated stochastic matrix
2. Add self loops to each node (optional)
3. Normalize the matrix
4. Expand by taking the *e*th power of the matrix **(random walk)**
5. Inflate by taking inflation of the resulting matrix with parameter *r*
6. Repeat steps 4 and 5 until a steady state is reached (convergence).

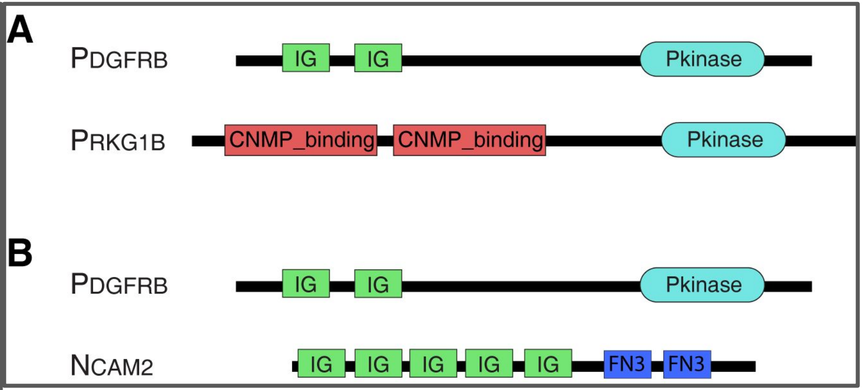
Remark. A simple principle, but that depends heavily on the inflation parameter *r*.



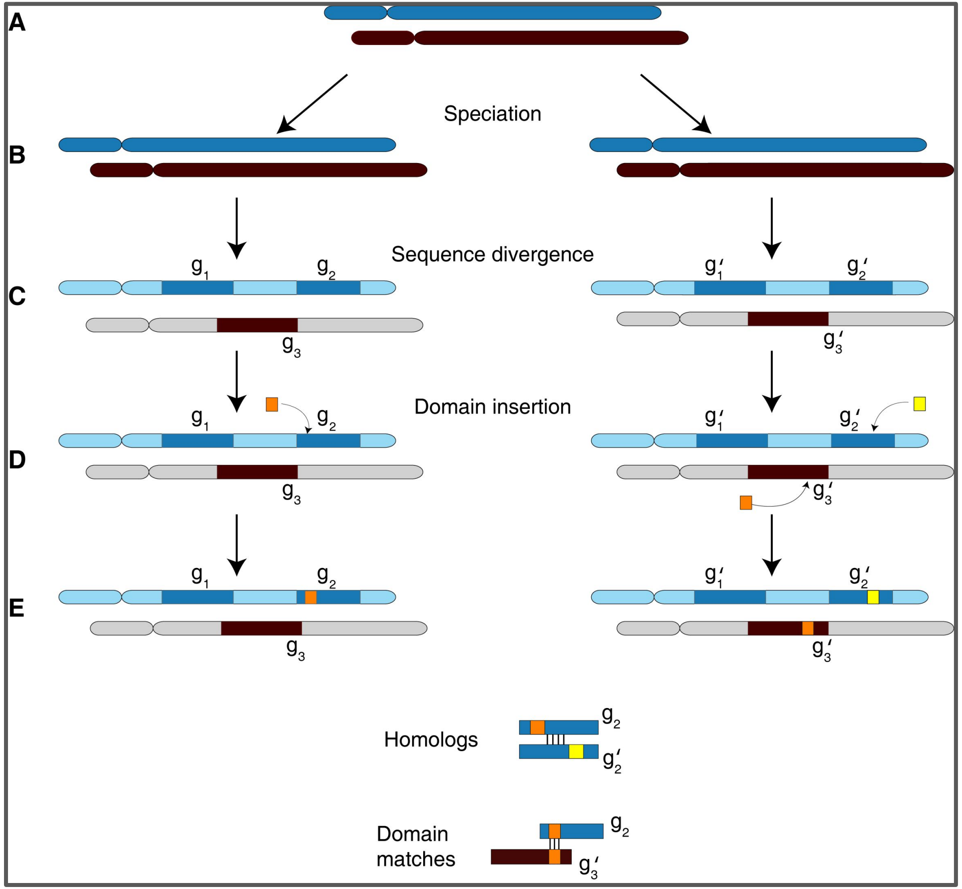
Neighborhood correlation (Song et al, 2008)

Motivation: Multi-domain proteins are ubiquitous, and are quite prone to domain insertion (non-vertical evolution), that can confuse sequence similarity.





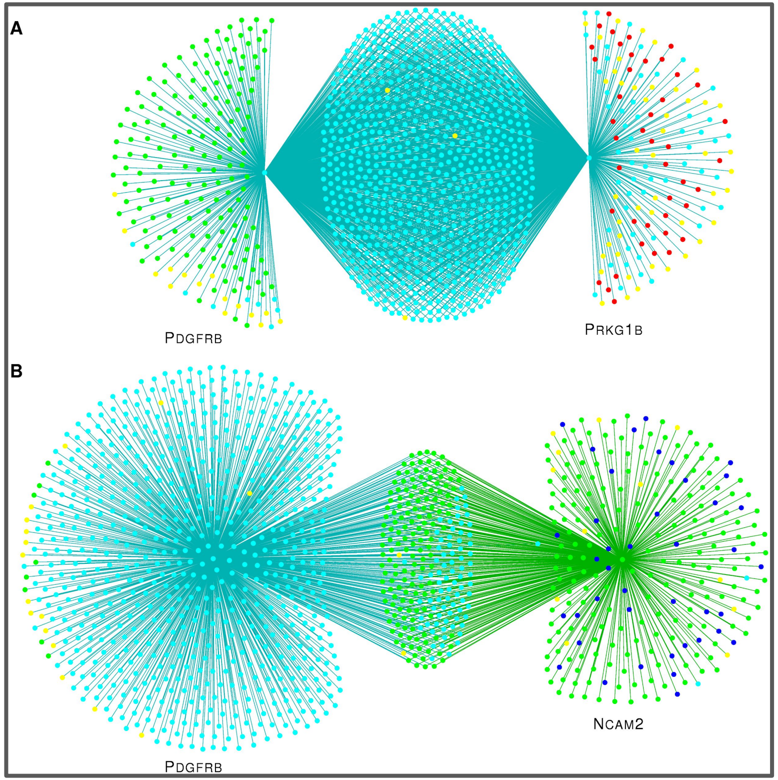
Homology versus domain-match

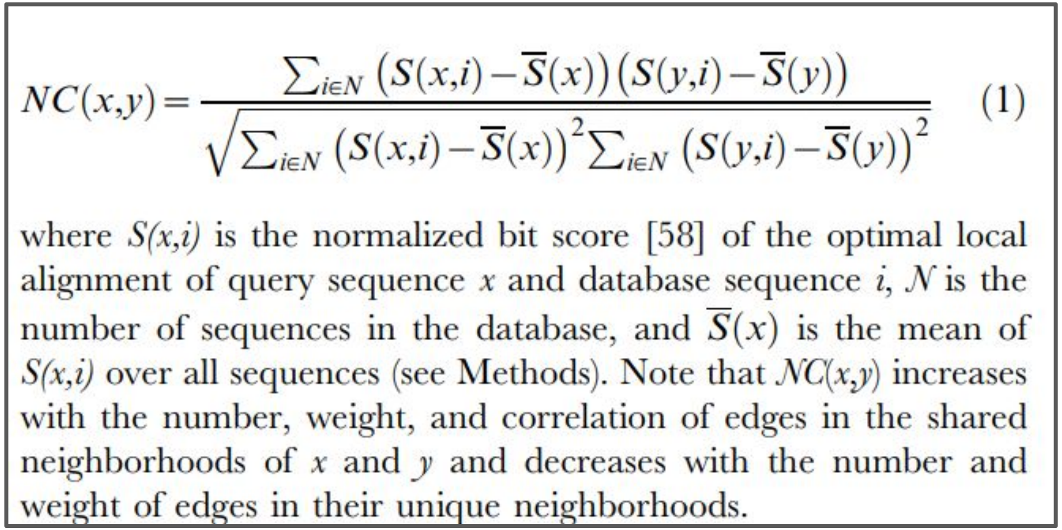


Despite comparable sequence similarity, the genomic context (g1 and g1’are clear homologs) suggests that g2 and g2’ are homologous, while g2 and g3’ are not.

The Neighborhood Correlation (NC) score

Idea: In the sequence similarity graph, true homolog pairs will share a larger number of neighbours (genes that are from the same family) than domain-sharing pairs.





$$NC(x,y) = \frac{\sum_{i \in N} \left(S(x,i) - \overline{S}(x)\right)\left(S(y,i) - \overline{S}(y)\right)}{\sqrt{\sum_{i \in N} \left(S(x,i) - \overline{S}(x)\right)^2 \sum_{i \in N} \left(S(y,i) - \overline{S}(y)\right)^2}} \qquad (1)$$

where *S(x,i)* is the normalized bit score [58] of the optimal local alignment of query sequence *x* and database sequence *i*, *N* is the number of sequences in the database, and $\overline{S}(x)$ is the mean of *S(x,i)* over all sequences (see Methods). Note that *NC(x,y)* increases with the number, weight, and correlation of edges in the shared neighborhoods of *x* and *y* and decreases with the number and weight of edges in their unique neighborhoods.

Clustering: keep only edges of NC score >0.5 and take connected components.

Some results

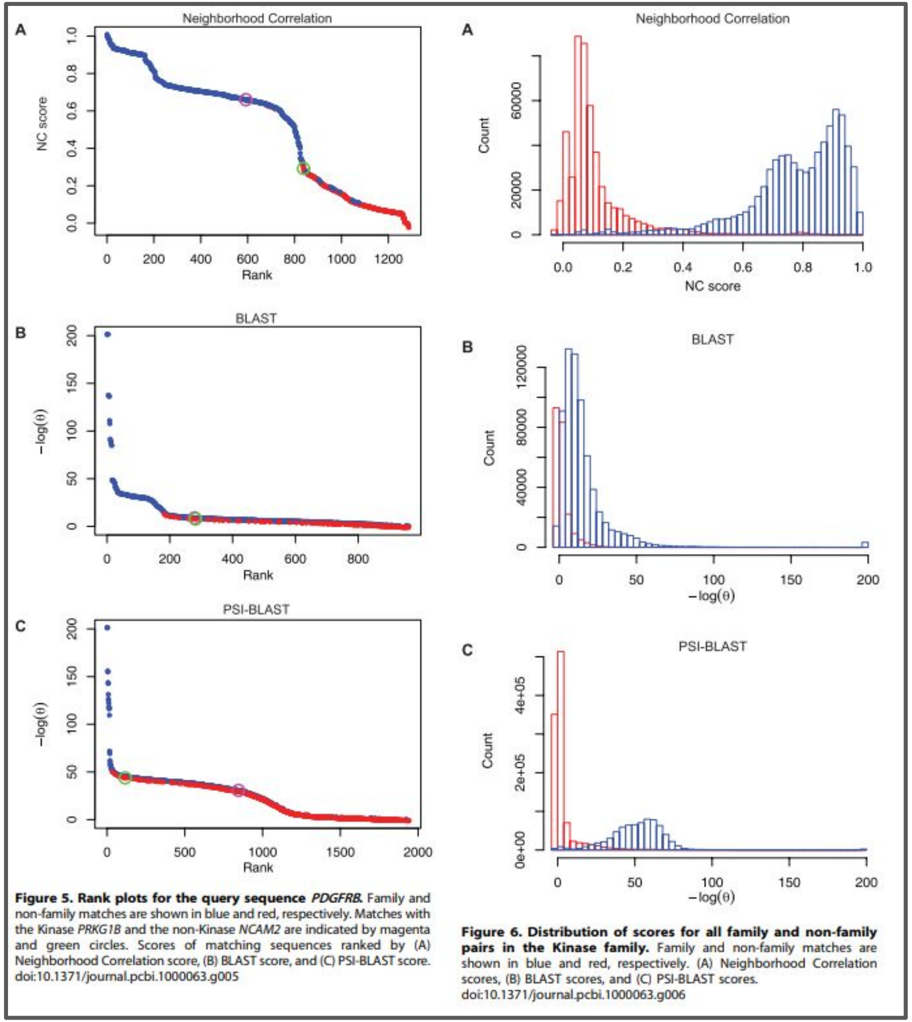
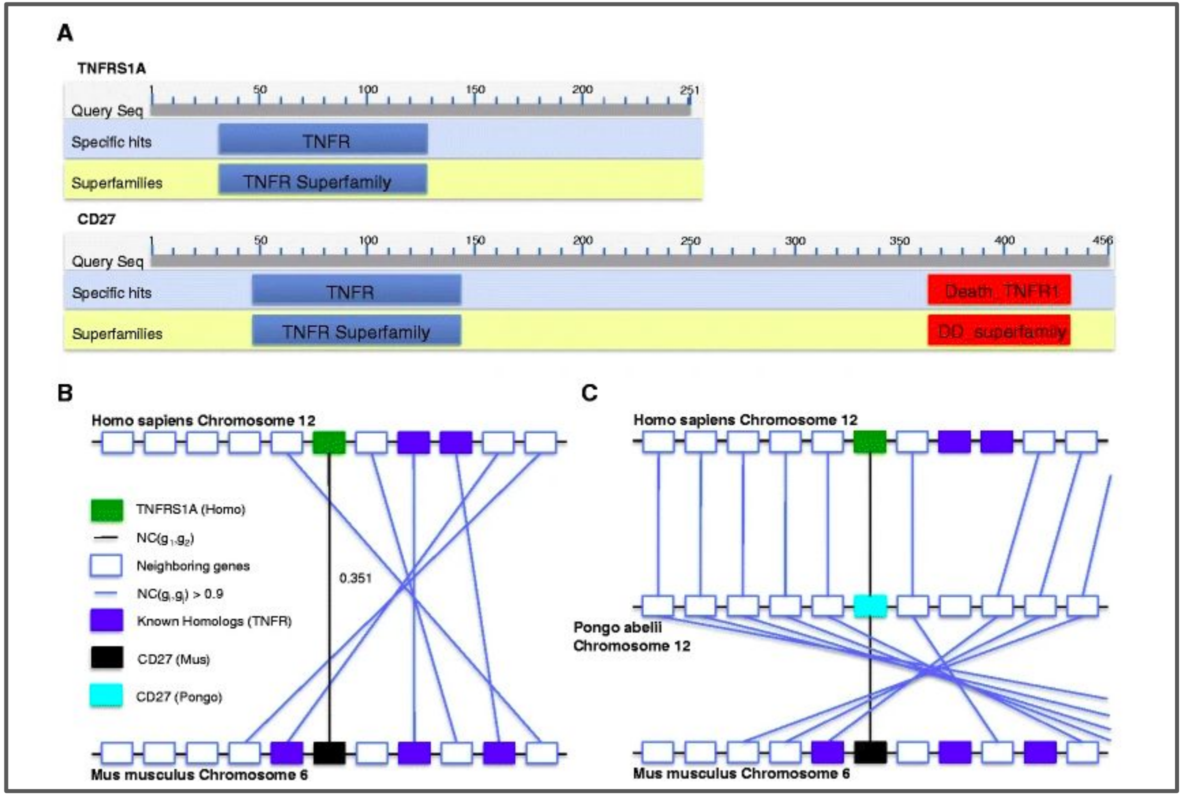


Figure 5. Rank plots for the query sequence *PDGFRB*. Family and non-family matches are shown in blue and red, respectively. Matches with the Kinase *PRKG1B* and the non-Kinase *NCAM2* are indicated by magenta and green circles. Scores of matching sequences ranked by (A) Neighborhood Correlation score, (B) BLAST score, and (C) PSI-BLAST score.
doi:10.1371/journal.pcbi.1000063.g005

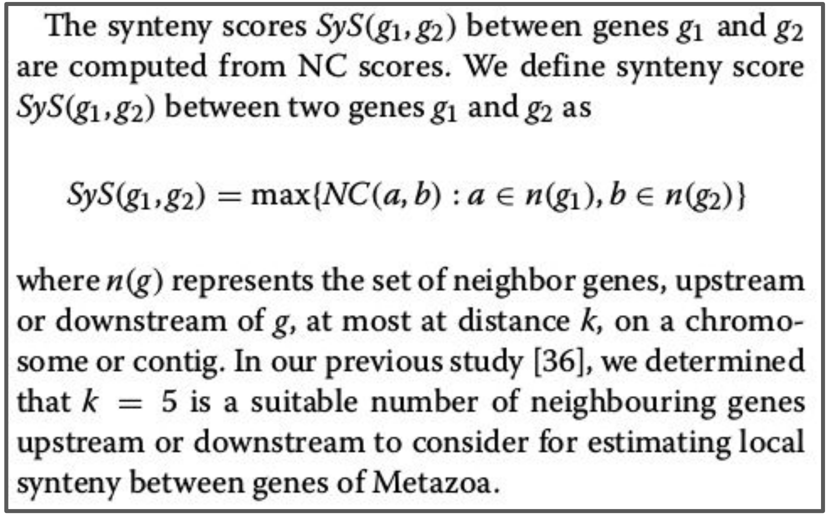
Figure 6. Distribution of scores for all family and non-family pairs in the Kinase family. Family and non-family matches are shown in blue and red, respectively. (A) Neighborhood Correlation scores, (B) BLAST scores, and (C) PSI-BLAST scores.
doi:10.1371/journal.pcbi.1000063.g006

GenFamClust (Ali et al, 2013, 2016)

Principle: The syntenic context around potential homologs strengthen the sequence similarity signal especially when it is close to the threshold accepted to define homologs.



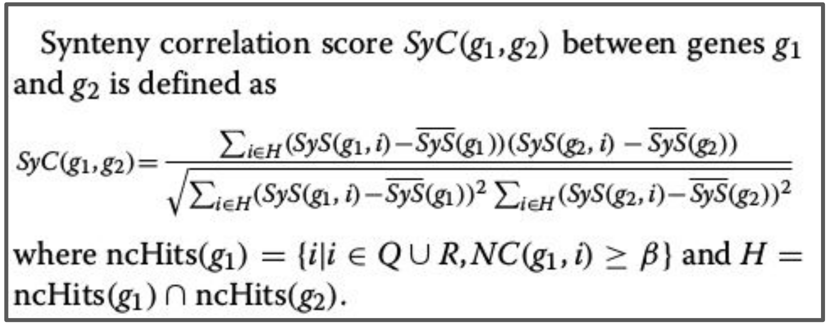
GenFamClust: synteny score and correlation



The synteny scores $SyS(g_1,g_2)$ between genes g_1 and g_2 are computed from NC scores. We define synteny score $SyS(g_1,g_2)$ between two genes g_1 and g_2 as

$$SyS(g_1,g_2) = \max\{NC(a,b) : a \in n(g_1), b \in n(g_2)\}$$

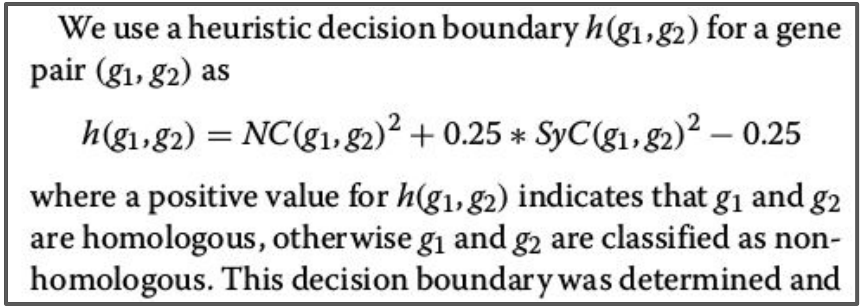
where $n(g)$ represents the set of neighbor genes, upstream or downstream of g, at most at distance k, on a chromosome or contig. In our previous study [36], we determined that $k = 5$ is a suitable number of neighbouring genes upstream or downstream to consider for estimating local synteny between genes of Metazoa.



Synteny correlation score $SyC(g_1,g_2)$ between genes g_1 and g_2 is defined as

$$SyC(g_1,g_2) = \frac{\sum_{i\in H}(SyS(g_1,i)-\overline{SyS}(g_1))(SyS(g_2,i)-\overline{SyS}(g_2))}{\sqrt{\sum_{i\in H}(SyS(g_1,i)-\overline{SyS}(g_1))^2 \sum_{i\in H}(SyS(g_2,i)-\overline{SyS}(g_2))^2}}$$

where $\text{ncHits}(g_1) = \{i | i \in Q \cup R, NC(g_1,i) \geq \beta\}$ and $H = \text{ncHits}(g_1) \cap \text{ncHits}(g_2)$.



We use a heuristic decision boundary $h(g_1,g_2)$ for a gene pair (g_1,g_2) as

$$h(g_1,g_2) = NC(g_1,g_2)^2 + 0.25 * SyC(g_1,g_2)^2 - 0.25$$

where a positive value for $h(g_1,g_2)$ indicates that g_1 and g_2 are homologous, otherwise g_1 and g_2 are classified as non-homologous. This decision boundary was determined and

R = reference dataset for which homology has already been computed (Q, query dataset if no reference is available).

Clustering is done using single linkage clustering.

SynFuse (Forteza, 2017): correcting gene families

Motivation:

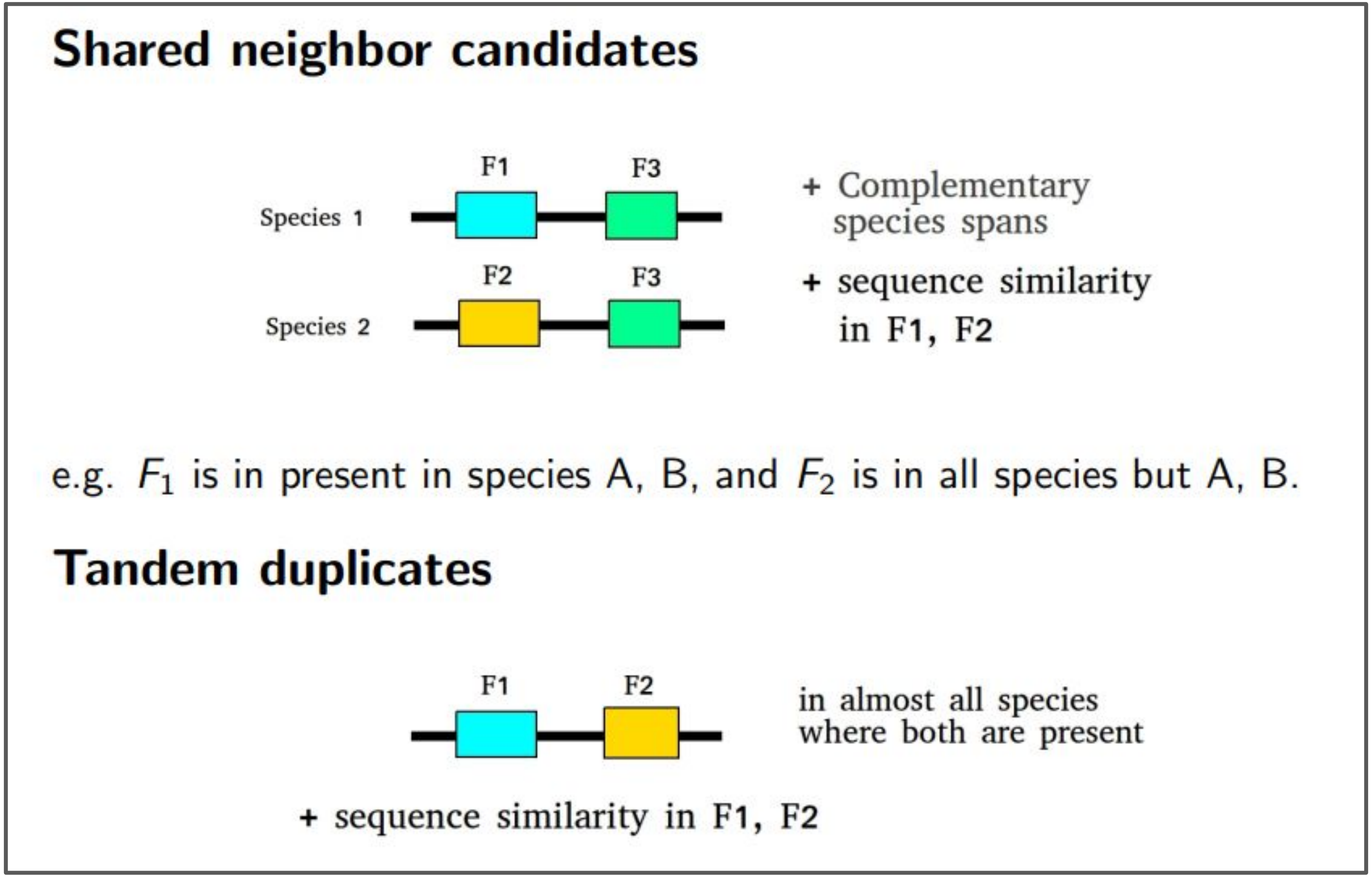
- All the methods for inferring homologous gene families rely essentially on parameterized clustering algorithms, applied to a graph where potential homology has been reduced to a single weight, with no post-processing of the results.
- On the *Anopheles* dataset, we can observe that there are many OrthoDB small families, that are likely the result of splitting true, larger, families.

Our goal is to design a method that will address this specific error in gene families results.

Principle:

- Identify candidate gene family pairs that show some signal that they might result from a wrong split of a single family, using synteny as the main signal, complemented by sequence similarity and phylogenetics.
- Design a *fusion score/test* that tells us if we accept to join both families into a single larger one.

SynFuse: candidates



Shared neighbor candidates

+ Complementary species spans

+ sequence similarity in F1, F2

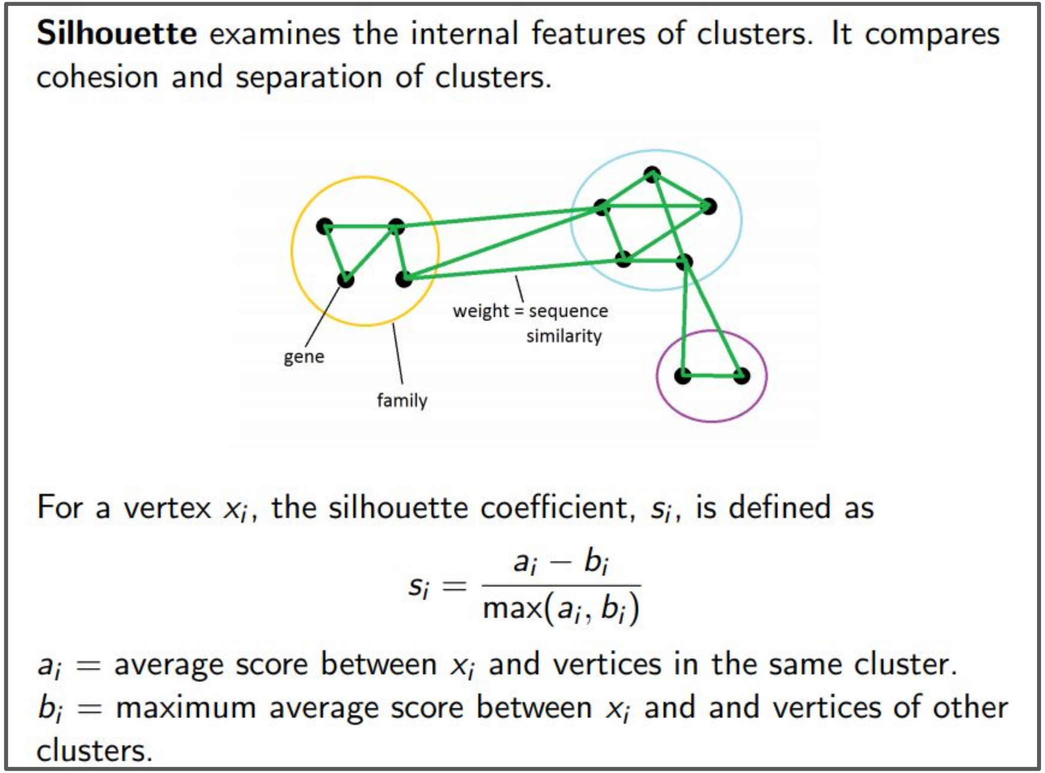
e.g. F_1 is in present in species A, B, and F_2 is in all species but A, B.

Tandem duplicates

in almost all species where both are present

+ sequence similarity in F1, F2

SynFuse: fusion score



Silhouette examines the internal features of clusters. It compares cohesion and separation of clusters.

For a vertex x_i, the silhouette coefficient, s_i, is defined as

$$s_i = \frac{a_i - b_i}{\max(a_i, b_i)}$$

a_i = average score between x_i and vertices in the same cluster.
b_i = maximum average score between x_i and and vertices of other clusters.

For a candidate pair (F1,F2), compute

- the sum of the silhouettes of genes in F1 and in F2, divided by |F1|+|F2|, denoted by *s*,

- the weighted average silhouette of the genes in F1 U F2, denoted by *s12*.

FusionScore(F1,F2) = *s12 - s*

SynFuse: fusion test

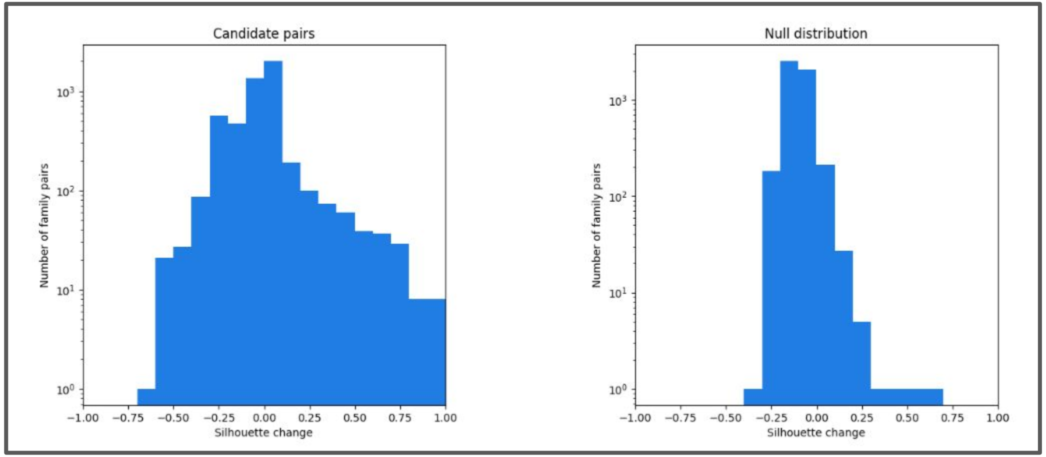
Now, we have a fusion score for a given candidate.
How do-we decide if this fusion score is good enough to join F1 and F2?

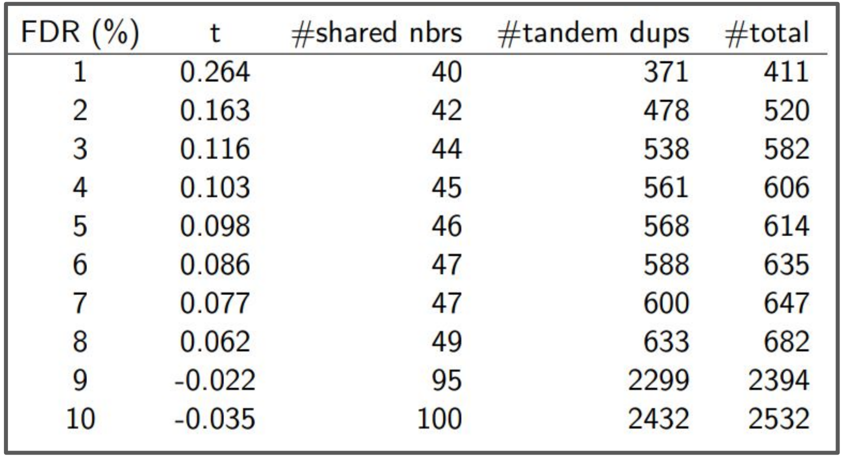
As we consider many candidate pairs, we face the issue of **Multiple Hypothesis Tests**: if we only set a threshold, we know that by considering an increasing number of candidate pairs, we will find some to fuse (False Positives, FP). So we want to control this rate of FP.

We use a **False Discovery Rate (FDR)** approach:
FDR gives the probability that a score higher than some threshold can be found randomly under a provided **null hypothesis distribution**.
For a threshold *t*, define *s_b* to be the number of observed data (silhouette change in candidate pairs) with a score above *t*, and *s_n* to be the number of null data with higher scores than *t*. We define FDR = *s_n /s_b*.

Null hypothesis: shuffle inter-clusters edges and shuffle intra-clusters edge weights.

SynFuse: results

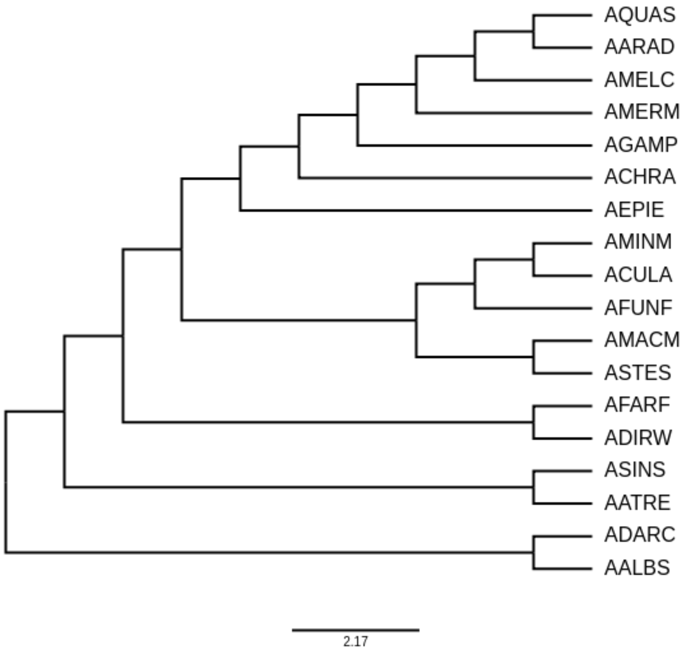




FDR (%)	t	#shared nbrs	#tandem dups	#total
1	0.264	40	371	411
2	0.163	42	478	520
3	0.116	44	538	582
4	0.103	45	561	606
5	0.098	46	568	614
6	0.086	47	588	635
7	0.077	47	600	647
8	0.062	49	633	682
9	-0.022	95	2299	2394
10	-0.035	100	2432	2532

SynFuse: example MZ22528769,MZ22508297

ACULA	AXCM01016875	MZ22528769
ACULA	KI422741	MZ22528769
AFUNF	KB669281	MZ22528769
AALBS	KB672435	MZ22508297
AARAD	KB704596	MZ22508297
AATRE	KI421900	MZ22508297
ACHRA	KB680646	MZ22508297
ADARC	scaffold_224	MZ22508297
ADIRW	KB672880	MZ22508297
AEPIE	KB670636	MZ22508297
AFARF	KI421558	MZ22508297
AGAMP	3R	MZ22508297
AMACU	KI433519	MZ22508297
AMINM	KB663722	MZ22508297
AQUAS	KB666365	MZ22508297
ASINS	KI397843	MZ22508297
ASTES	KB664514	MZ22508297



SynFuse: example

ACULA	KI422741	**MZ22528769**	ACUA021098	-	**10**	11927
ACULA	KI422741	**MZ22518448**	ACUA013350	+	30670	31553
AFUNF	KB669281	**MZ22518448**	AFUN003124	-	53423	54308
AFUNF	KB669281	**MZ22528769**	AFUN003125	+	79730	89714

AARAD	KB704596	**MZ22518448**	AARA014622	-	1602845	1603717
AARAD	KB704596	**MZ22508297**	AARA007168	+	1606383	1653782
AATRE	KI421900	**MZ22518448**	AATE002235	-	778705	779590
AATRE	KI421900	**MZ22508297**	AATE000430	+	801980	803622

...

Sequence similarity (NC score):

AARA007168	ACUA021098	0.30063245

…

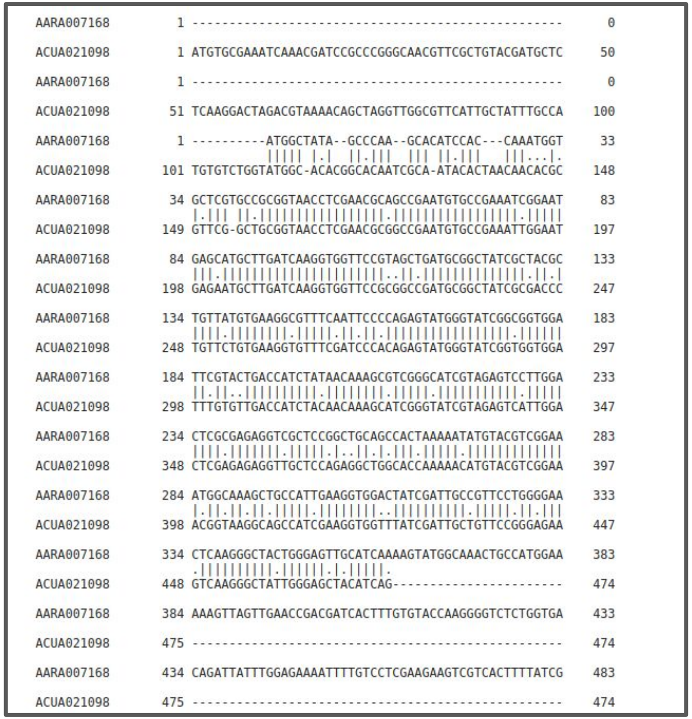
Silhouette improvement:

#F1	F2	edges	s1	s2	s12	s	d
MZ22528769	MZ22508297	20	0.611647	0.217636	0.920092	0.287167	0.632925

SynFuse: example

ACULA	KI422741	MZ22528769	ACUA021098	-	10	11927
AARAD	KB704596	MZ22508297	AARA007168	+	1606383	1653782

Sequence similarity (NC score): AARA007168 ACUA021098 0.30063245



```
AARA007168        1 --------------------------------------------------      0

ACUA021098        1 ATGTGCGAAATCAAACGATCCGCCCGGGCAACGTTCGCTGTACGATGCTC     50

AARA007168        1 --------------------------------------------------      0

ACUA021098       51 TCAAGGACTAGACGTAAAACAGCTAGGTTGGCGTTCATTGCTATTTGCCA    100

AARA007168        1 ----------ATGGCTATA--GCCCAA--GCACATCCAC---CAAATGGT     33
                              ||||| |.|  ||.|||  ||| ||.|||   |||...|.
ACUA021098      101 TGTGTCTGGTATGGC-ACACGGCACAATCGCA-ATACACTAACAACACGC    148

AARA007168       34 GCTCGTGCCGCGGTAACCTCGAACGCAGCCGAATGTGCCGAAATCGGAAT     83
                    |.||| ||.|||||||||||||||||.|||||||||||||||||.|||||
ACUA021098      149 GTTCG-GCTGCGGTAACCTCGAACGCGGCCGAATGTGCCGAAATTGGAAT    197

AARA007168       84 GAGCATGCTTGATCAAGGTGGTTCCGTAGCTGATGCGGCTATCGCTACGC    133
                    |||.||||||||||||||||||||||..||.||||||||||||||.||.|
ACUA021098      198 GAGAATGCTTGATCAAGGTGGTTCCGCGGCCGATGCGGCTATCGCGACCC    247

AARA007168      134 TGTTATGTGAAGGCGTTTCAATTCCCCAGAGTATGGGTATCGGCGGTGGA    183
                    ||||.||||||||.|||||.||.||.|||||||||||||||||.||||||
ACUA021098      248 TGTTCTGTGAAGGTGTTTCGATCCCACAGAGTATGGGTATCGGTGGTGGA    297

AARA007168      184 TTCGTACTGACCATCTATAACAAAGCGTCGGGCATCGTAGAGTCCTTGGA    233
                    ||.||..||||||||||.||||||||.|||||.|||||||||||.|||||
ACUA021098      298 TTTGTGTTGACCATCTACAACAAAGCATCGGGTATCGTAGAGTCATTGGA    347

AARA007168      234 CTCGCGAGAGGTCGCTCCGGCTGCAGCCACTAAAAATATGTACGTCGGAA    283
                    ||||.|||||||.|||||.|..||.|.|||.|||||.|||||||||||||
ACUA021098      348 CTCGAGAGAGGTTGCTCCAGAGGCTGGCACCAAAAACATGTACGTCGGAA    397

AARA007168      284 ATGGCAAAGCTGCCATTGAAGGTGGACTATCGATTGCCGTTCCTGGGGAA    333
                    |.||.||.||.|||||.||||||||..||||||||||.|||||.||.|||
ACUA021098      398 ACGGTAAGGCAGCCATCGAAGGTGGTTTATCGATTGCTGTTCCGGGAGAA    447

AARA007168      334 CTCAAGGGCTACTGGGAGTTGCATCAAAAGTATGGCAAACTGCCATGGAA    383
                    .||||||||||.||||||.|.|||||.
ACUA021098      448 GTCAAGGGCTATTGGGAGCTACATCAG-----------------------    474

AARA007168      384 AAAGTTAGTTGAACCGACGATCACTTTGTGTACCAAGGGGTCTCTGGTGA    433

ACUA021098      475 --------------------------------------------------    474

AARA007168      434 CAGATTATTTGGAGAAAATTTTGTCCTCGAAGAAGTCGTCACTTTTATCG    483

ACUA021098      475 --------------------------------------------------    474
```

The other ACULA gene from family MZ22528769 is at the end of a short single-gene contig.

We can make the hypothesis that in both cases, the suffix of the gene was not assembled, resulting in clustering the incomplete genes as a separate family.

Question: could-we use this “signal” to try to improve the assembly by finding some contigs whose extremity contains the missing genes suffix?

SynFuse: potential?

The general principle to try to correct a set of homologous gene families using synteny, sequence similarity and phylogenetics to detect pairs is likely sound.

The statistical approach can be discussed. It might be an overkill to go through a large number of silhouette computations. Moreover, computing the thresholds linked to a chosen FDR requires also a lot of computation time.

The fusion of tandem duplicate families: is-it a good idea?

Actually, is the silhouette based approach the good one? Could we consider other (more local) features of a fused family that tells us it was good decision to fuse a candidate pair? For example how would the Multiple Sequence Alignment (MSA) of F1 U F2 would look like compared to the MSAs of F1 and F2?

Conclusion

- ❏ Gene family are defined in a multi-faceted way, including signal from sequence similarity, syntenic context, species coverage, phylogenetics.
- ❏ Most method rely on a pure clustering approach where a subset of these facets are abstracted into a single edge weight and are highly parameterized and threshold-based.
- ❏ There is likely room for improvement, especially for post-processing the clustering from one (or several) method(s) accounting for the signals for homology that have not been considered in the graph construction step.