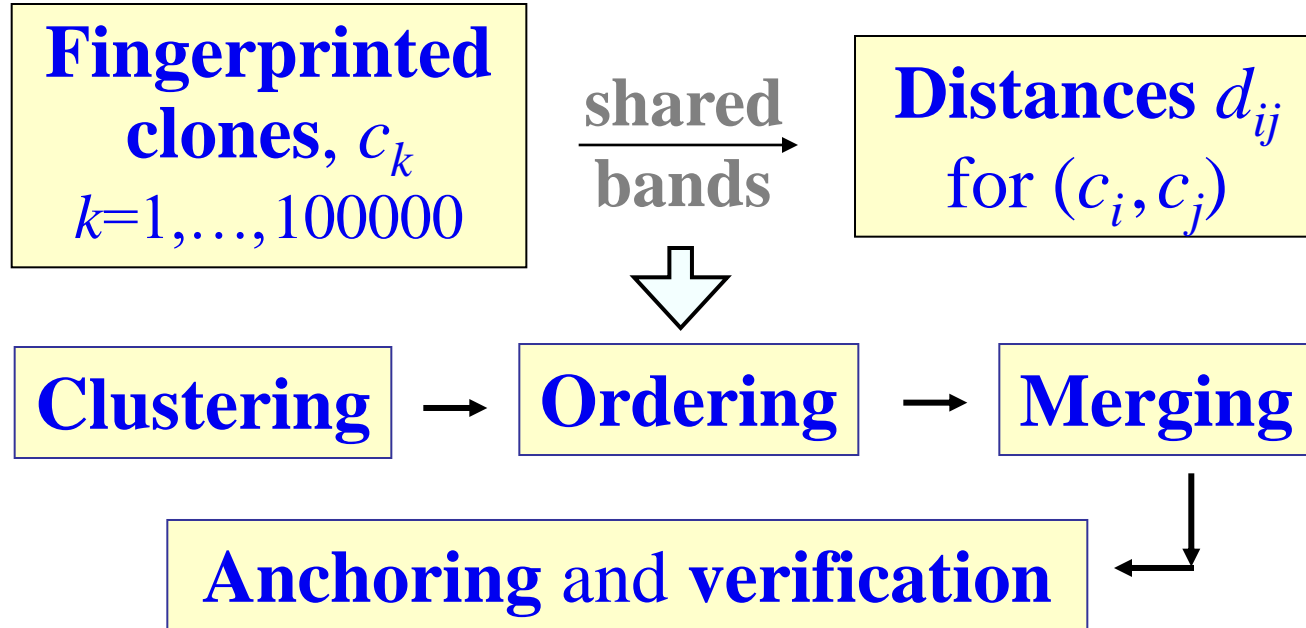


Using LTC software to assemble physical maps in complex genomes such as wheat

Zeev Frenkel, Etienne Paux, David Mester,
Catherine Feuillet, Abraham Korol



Major steps in physical mapping



A standard tool: FPC package

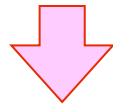
Genome mapping problems are computationally challenging

*“... We have been looking at the assemblies of large genomes ... and **for every ‘draft’ genome** we look at, we find hundreds - and sometimes thousands - of mis-assemblies”.*

Salzberg & Yorke (2005) Beware of mis-assembled genomes. *Bioinformatics*, **21**: 4320-4322

Some troubles

- Markers from non-neighbor chromosome regions may appear in one contig (due to chimerical clones)
- Adjacent clones from MTP sometimes fail to show overlapping → **gaps**
- Short contigs
- Contigs are sometimes *not linear*



Linear Topology Contigs - LTC software

LTC vs. FPC

- Adaptive clustering with liberal cutoffs
- Taking into account topological structure of the contigs
- More powerful methods for clone ordering
- A special way to deal with Q-clones
- ... other useful features

➔ Longer and more reliable contigs

Outline

Part 1 The approach and the algorithms

- LTC software: Goals and functions
- General logic and features of LTC
- Working with contigs: verification, elongation, merging, anchoring
- Testing FPC contigs by using LTC
- Some additional tools

Part 2 Implementation, demo, and examples

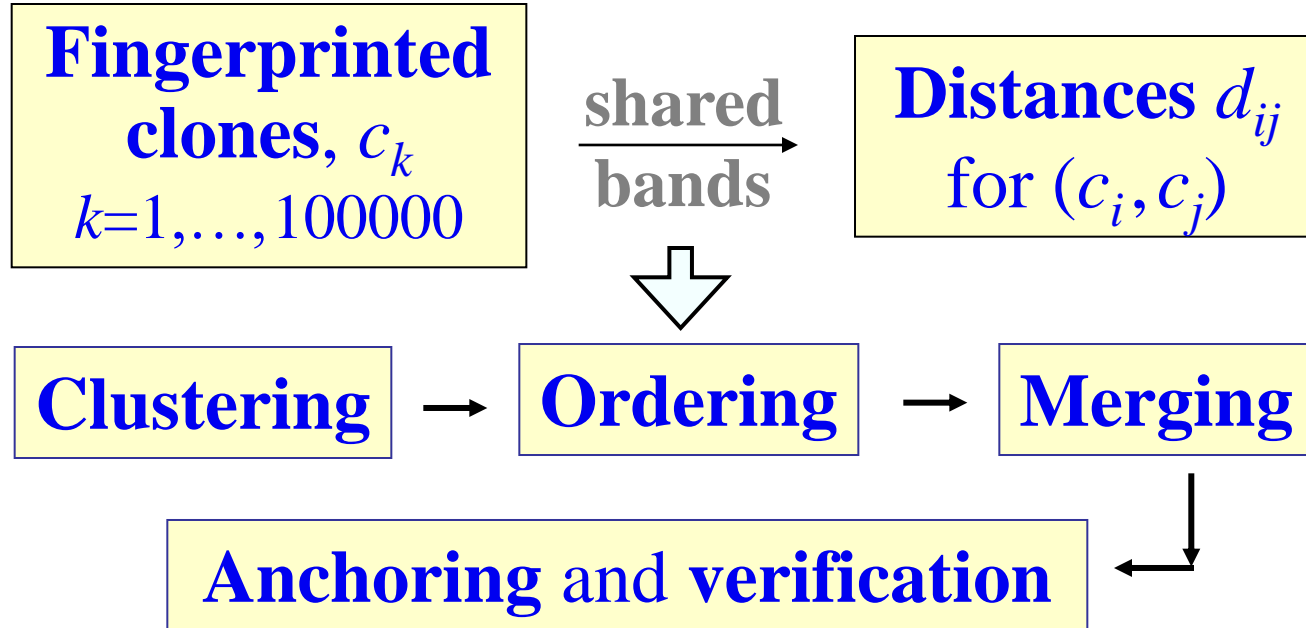
LTC: Goals and functions

The main goal: Robust contig assembly using BAC fingerprint data (HICF, STS markers, WGP tags)

LTC functions:

- Contig assembly, editing, verifying, and merging
- Curing gaps, reviewing, selection of alternative MTPs, reordering, and elongation of contigs obtained by other tools, e.g. by FPC package
- Tools for anchoring
- Simulating BAC fingerprint data

Major steps in physical mapping



A standard tool: FPC package

General logic and features of LTC

- Decision about clone overlapping based on more accurate estimates of p -values
- Adaptive clustering: increasing stringency
- Visualization of contig structure using ***network*** representation of clone overlaps
- Identification of Q-overlaps and Q-clones
- Breaking down clusters with *non-linear structure*

General logic and features of LTC

- Selection of MTP clones (without using band map)
- Estimation of clone-end coordinates
- Verification of clone position using jackknife resampling procedure
- Supercontig assembly coordinated with anchoring and synteny analysis

Elements of LTC

Frenkel et al. *BMC Bioinformatics* 2010, **11**:584
<http://www.biomedcentral.com/1471-2105/11/584>



METHODOLOGY ARTICLE

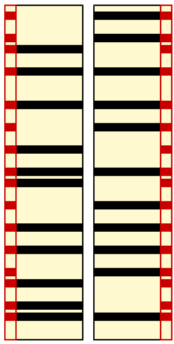
Open Access

LTC: a novel algorithm to improve the efficiency of contig assembly for physical mapping in complex genomes

Zeev Frenkel^{1*}, Etienne Paux², David Mester¹, Catherine Feuillet², Abraham Korol¹



P-value of clone overlaps



Sulston score (Sulston *et al.*, 1988):

$$S(c1, c2) = \sum_{k=n(c1 \cap c2)}^{n(c1)} \binom{n(c1)}{k} p^k (1-p)^{n(c1)-k}$$

$p = 1 - (1 - 1/N)^{n(c2)}$ is the probability of random coincidence of two bands;

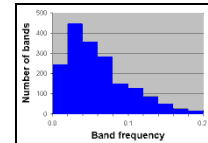
$n(c)$ – number of bands in clone c ;

N – total number of distinguishable bands

P-value of clone overlaps

Ways to improve:

- More accurate calculation of p -value using the same “random clones” model (e.g. Wendl, 2005)
- Taking into account common markers
- Taking into account band frequencies



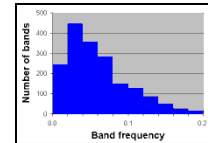
Consequences:

- Less false significant clone overlaps
- Possibility to use more liberal cutoffs
- By reducing the proportion of false overlaps more true overlaps can be considered

P-value of clone overlaps

Ways to improve:

- More accurate calculation of *p*-value using the same “random clones” model (e.g. Wendl, 2005)
- Taking into account common markers
- Taking into account band frequencies

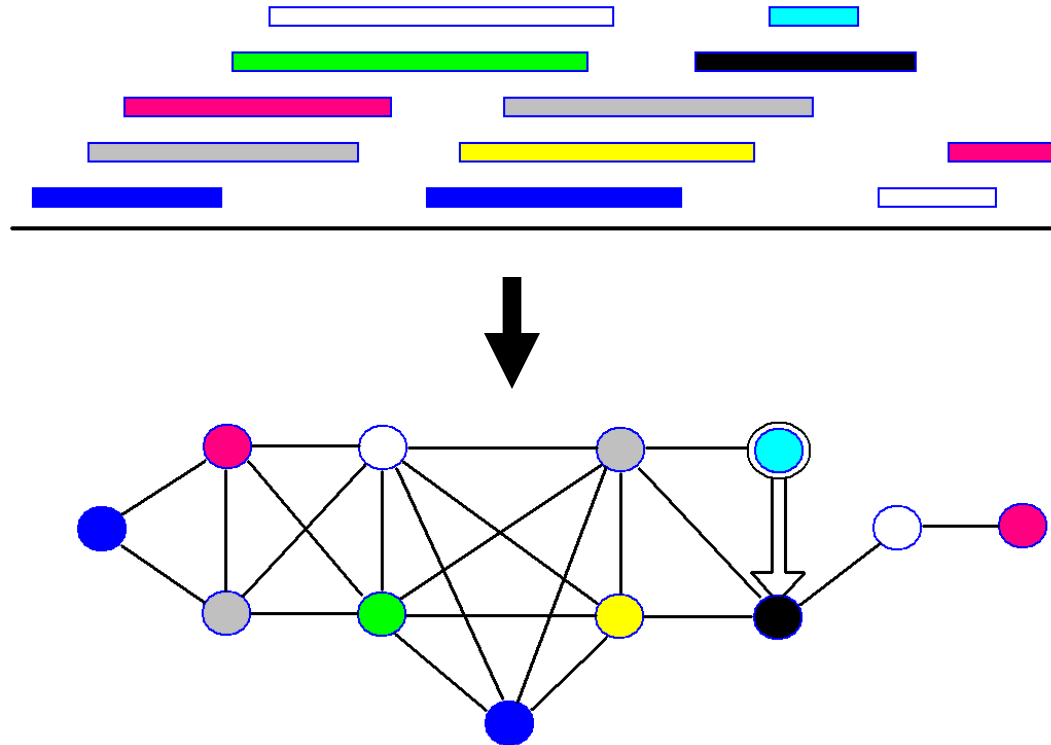


Consequences

- Less false overlaps
- Possibility to consider more true overlaps
- By reducing the number of false overlaps more true overlaps can be considered

We are using here the Sulston score just to allow full compatibility with standard FPC

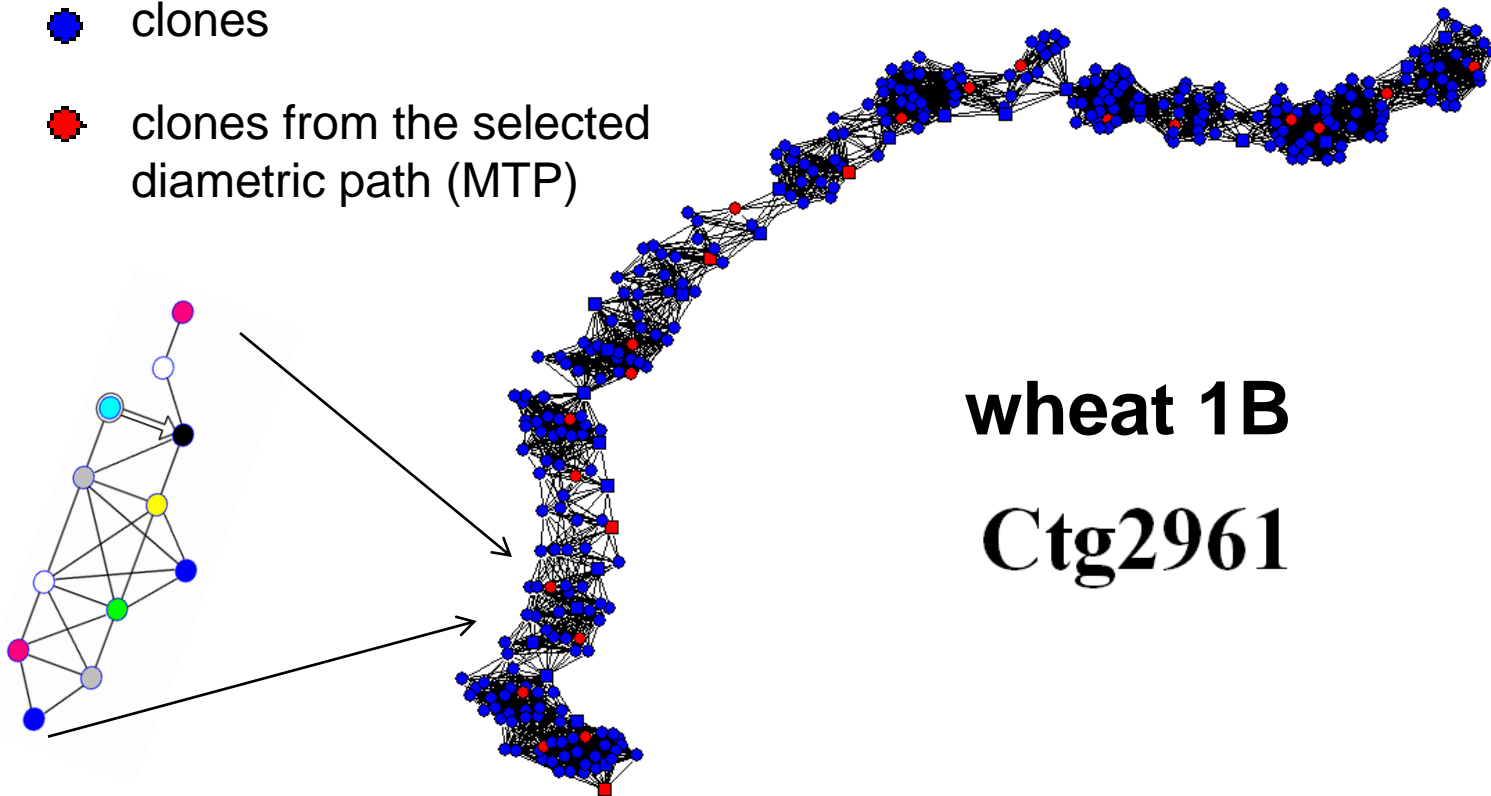
Network representation of significant clone overlaps



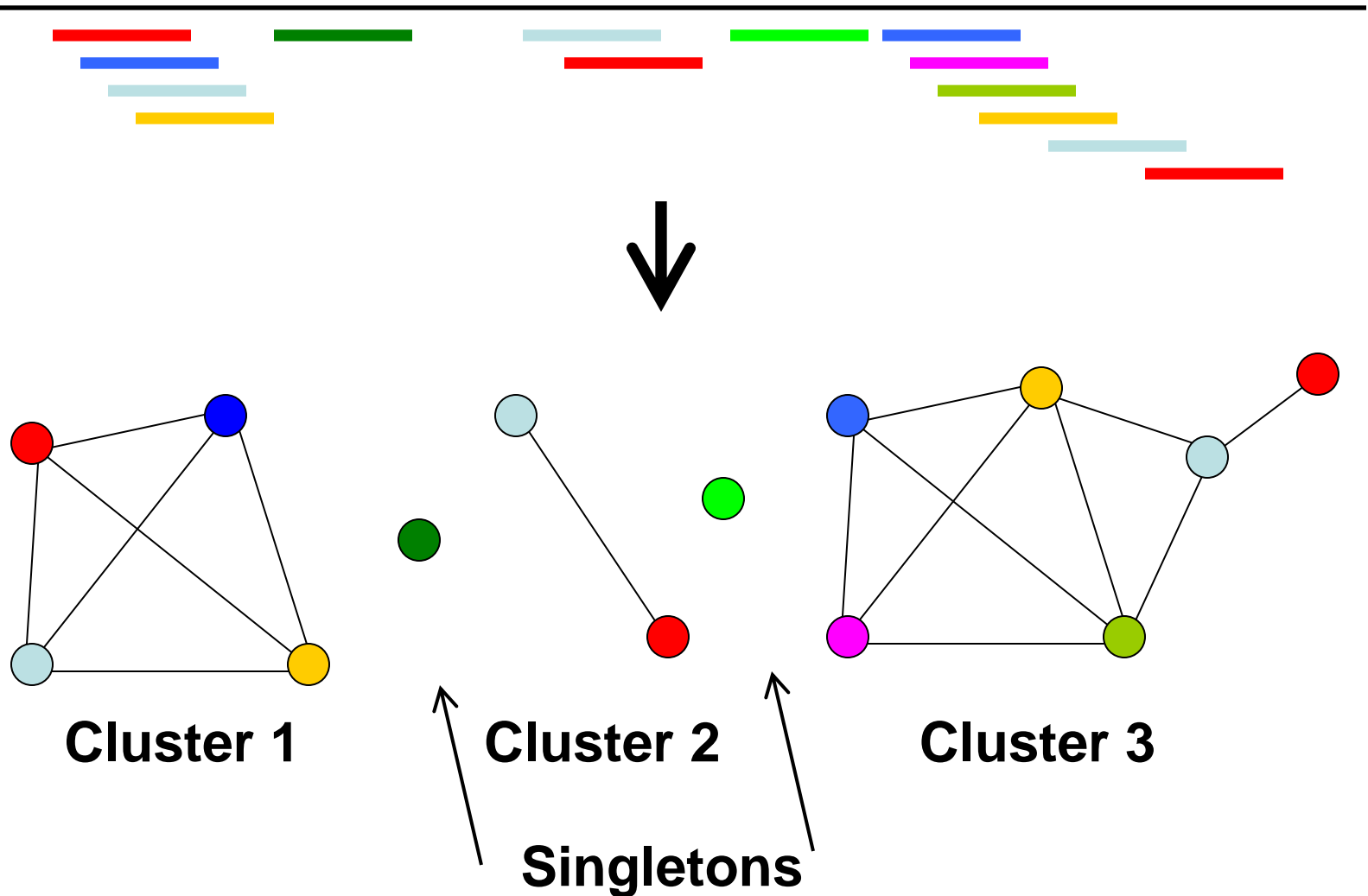
vertices correspond to clones and
edges – to significant clone overlaps

Network representation of significant clone overlaps

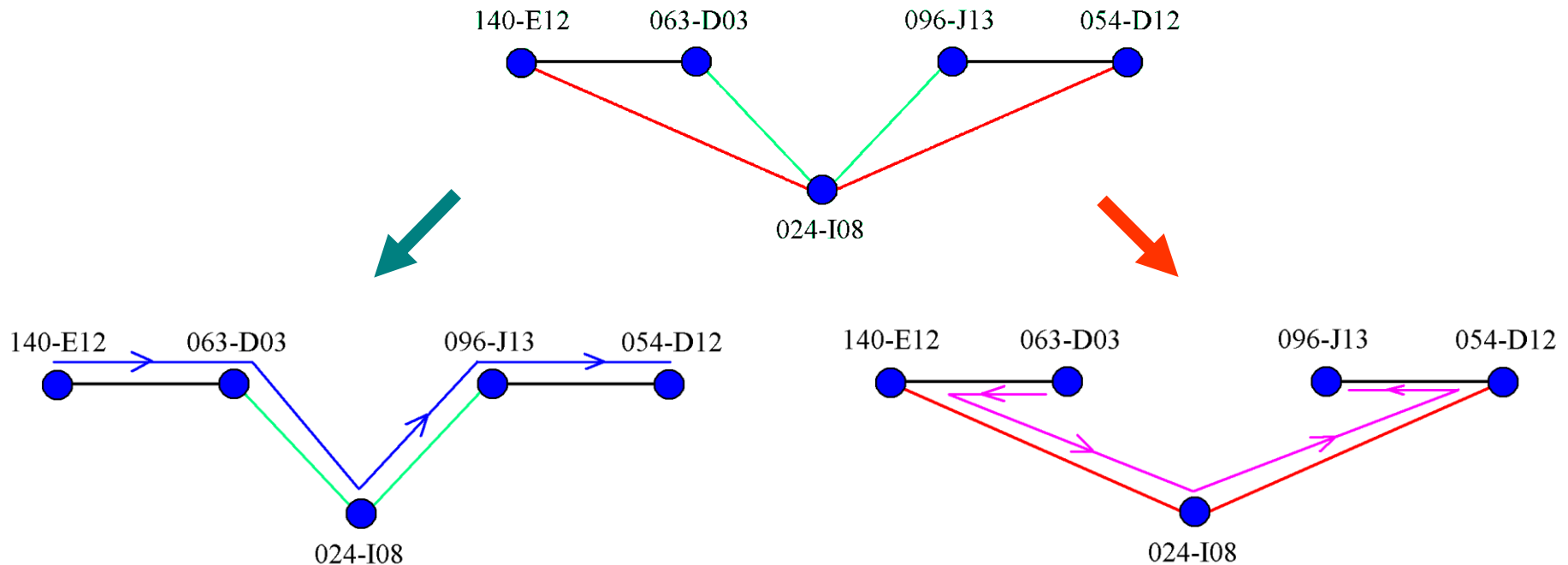
- clones
- clones from the selected
diametric path (MTP)



Net connectivity and clustering



Metrics matter: Different metrics → different cluster structure → different orders in MTP



Edges represent significant overlaps in corresponding metrics

— Lnn
— Sulston
— both (IoE)

Reasons for using *moderate size* clusters

Large clusters:

- Putatively chimerical?
- Difficult to analyze (e.g., building optimal band map)
- Variation of coverage along the chromosome
- Variation of repetitiveness

Short clusters:

- Laborious and uncertain elongation
- Laborious and less accurate anchoring

Obtaining moderate size clusters

Selecting p -value (cutoff) for clustering:

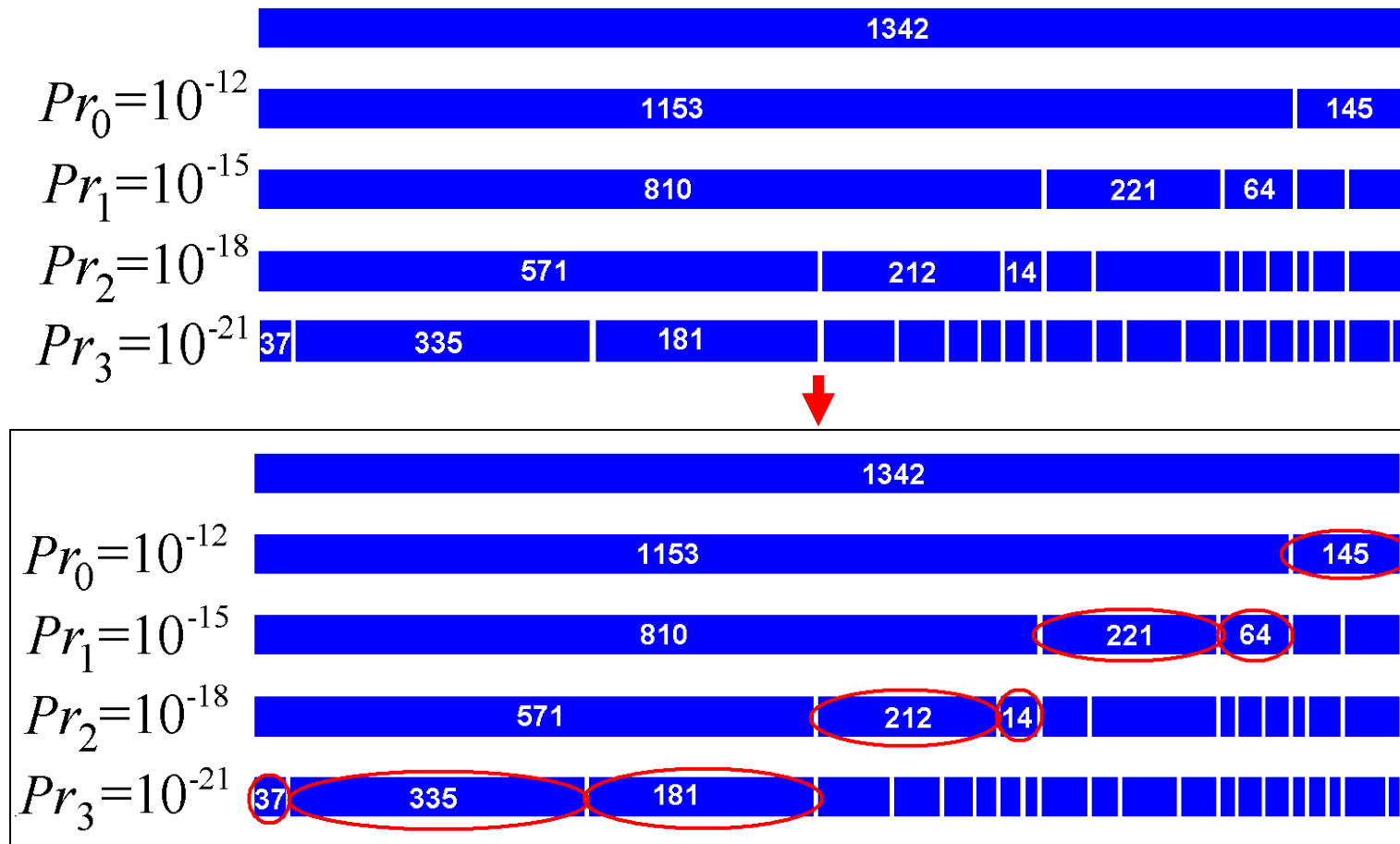
- Number of clones (multiple comparison problem)
- Observed distribution of shared band number (depends on coverage)

Excluding putatively problematic “bridges”:

- False significant overlaps
- Chimerical clones

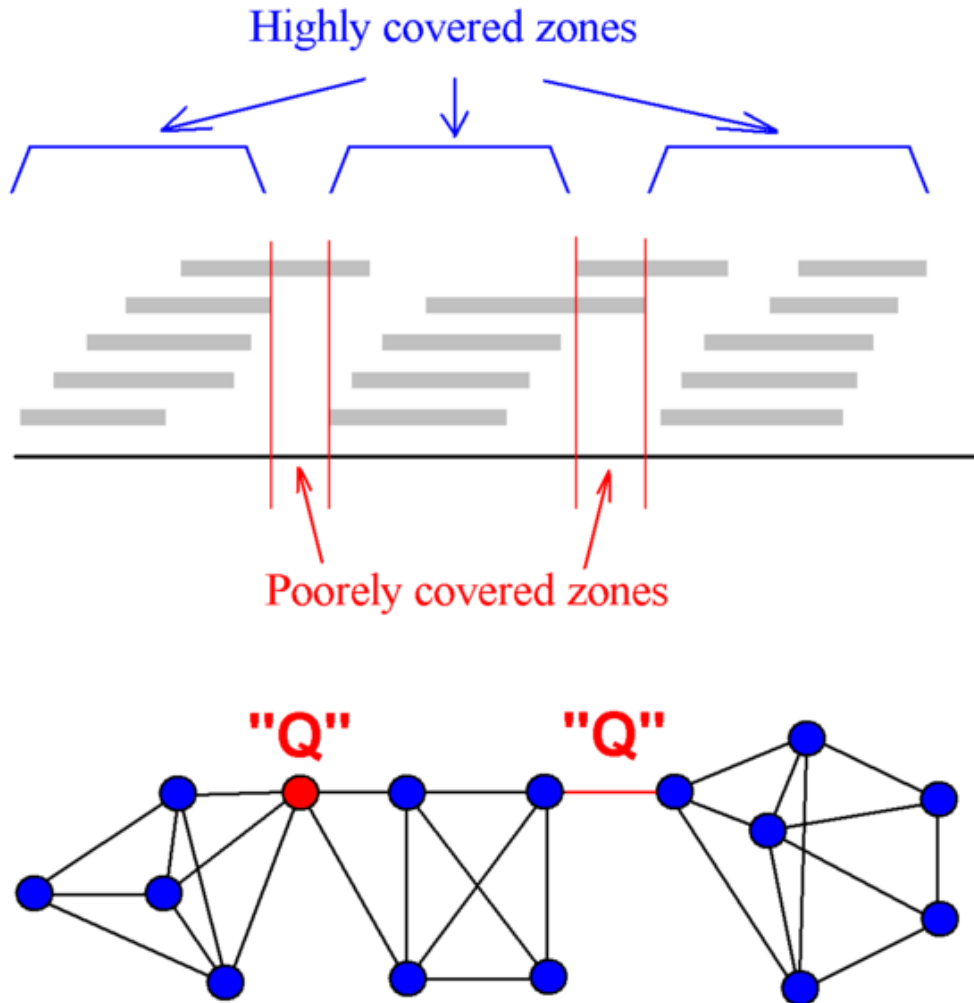
Adaptive Clustering

Varying cutoff: increasing rather than decreasing stringency



protecting “reasonable size” clusters

Putative Q-clones and Q-overlaps



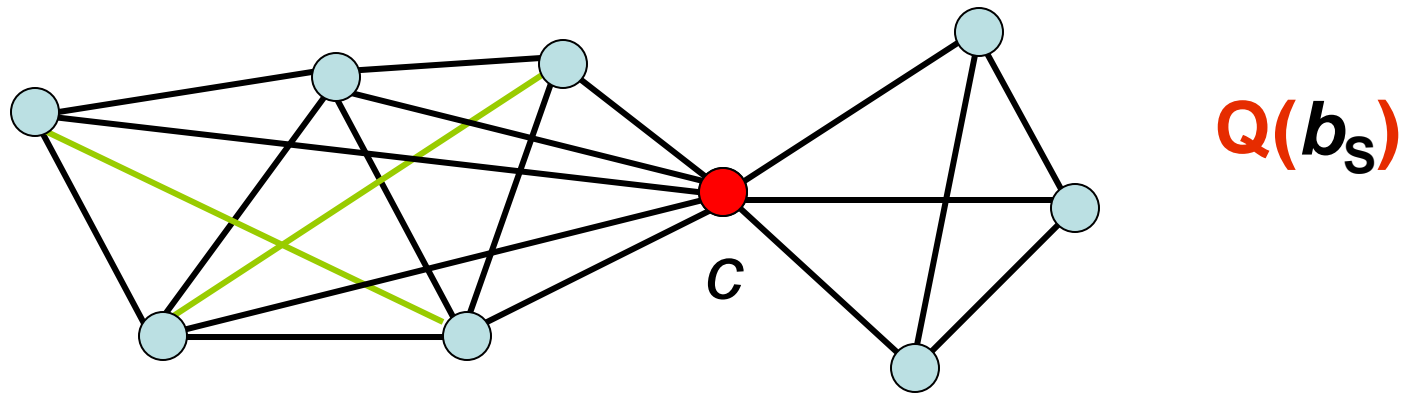
To address this problem we use an *additional, more liberal cutoff* →

Detecting Q-clones by using two cutoffs

Stringent cutoff $\mathbf{b_s}$ (say, 10^{-25})

A more liberal cutoff $\mathbf{b_L}$ (say, 10^{-15})

For each clone c :



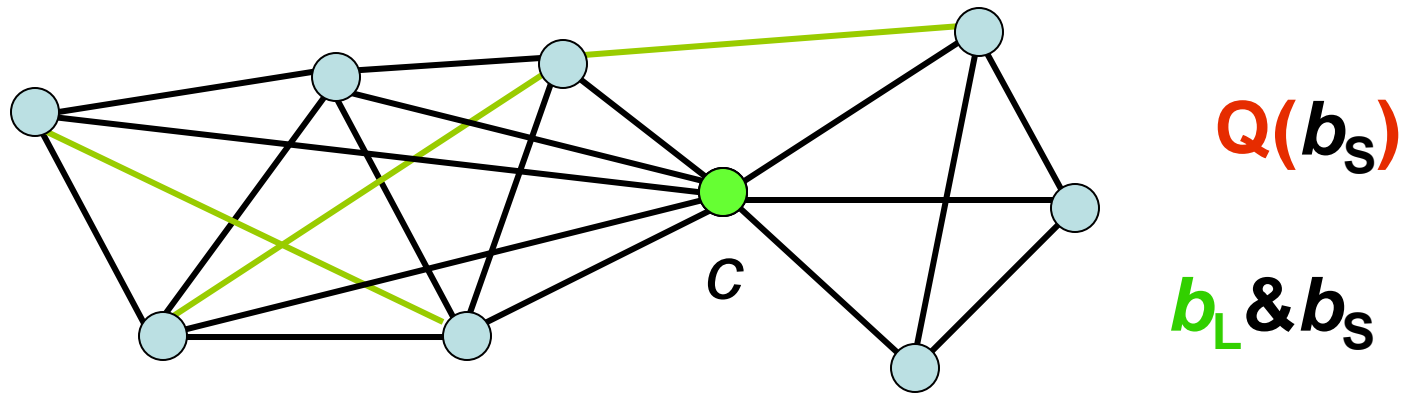
Q-clone $\mathbf{=: Q(b_L \& b_s)}$

Detecting Q-clones by using two cutoffs

Stringent cutoff $\mathbf{b_s}$ (say, 10^{-25})

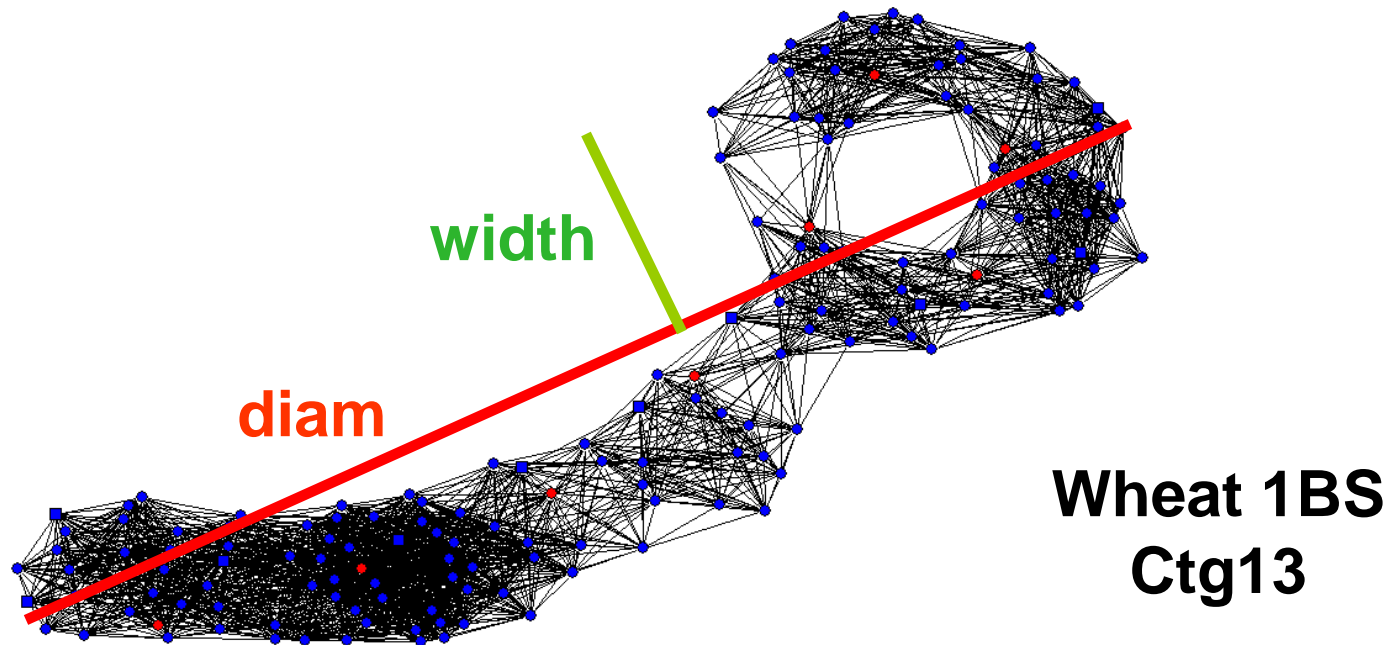
A more liberal cutoff $\mathbf{b_L}$ (say, 10^{-15})

For each clone c :



Identification of contig non-linearity

Using **net of significant clone overlaps** to find **diametric path** and calculate **width** of the net

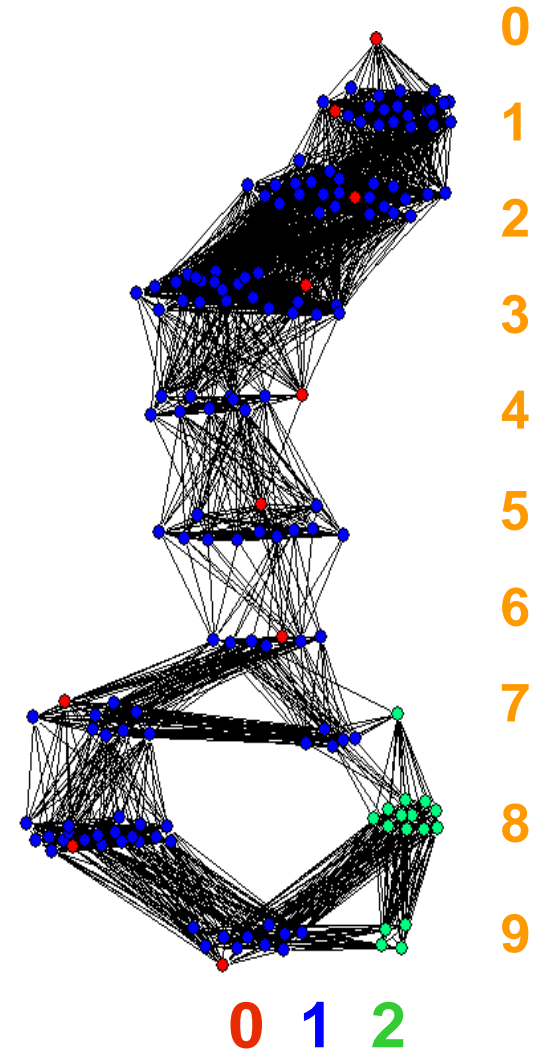


Width >1 is diagnostic for a **non-linear** cluster

Identification of contig non-linearity

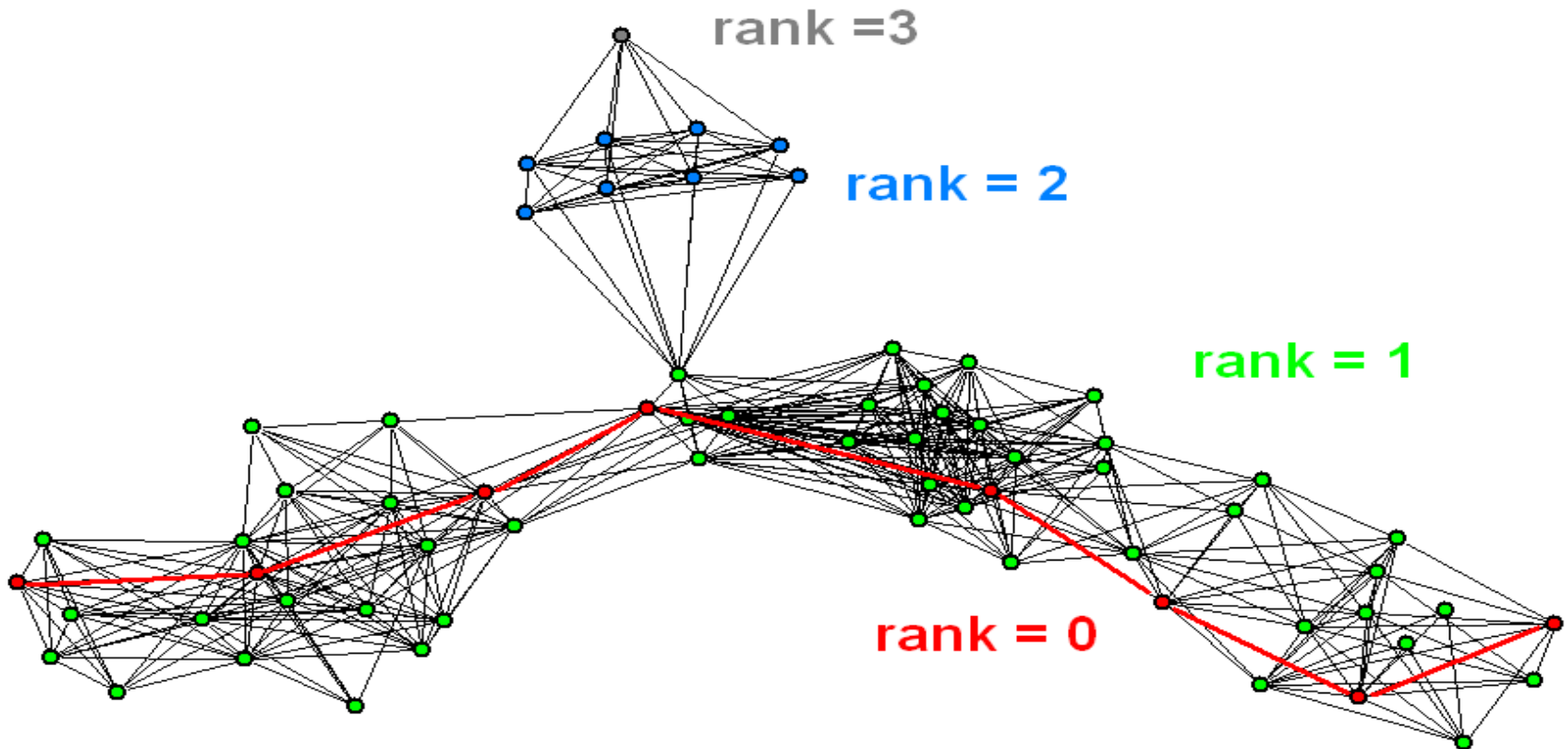
Diametric path:

- Calculate **ranks** $r_j = r_j(c_i)$ for all clones c_j relative to clone c_i (through significant clone overlaps).
- **Diametric path** (\rightarrow MTP) is the shortest path through significant clone overlaps connecting clones c_i and c_j with maximal $r_j(c_i)$.
- **Width of net**: maximal rank relative to **diametric path**
- **Width > 1 \rightarrow non-linear cluster**



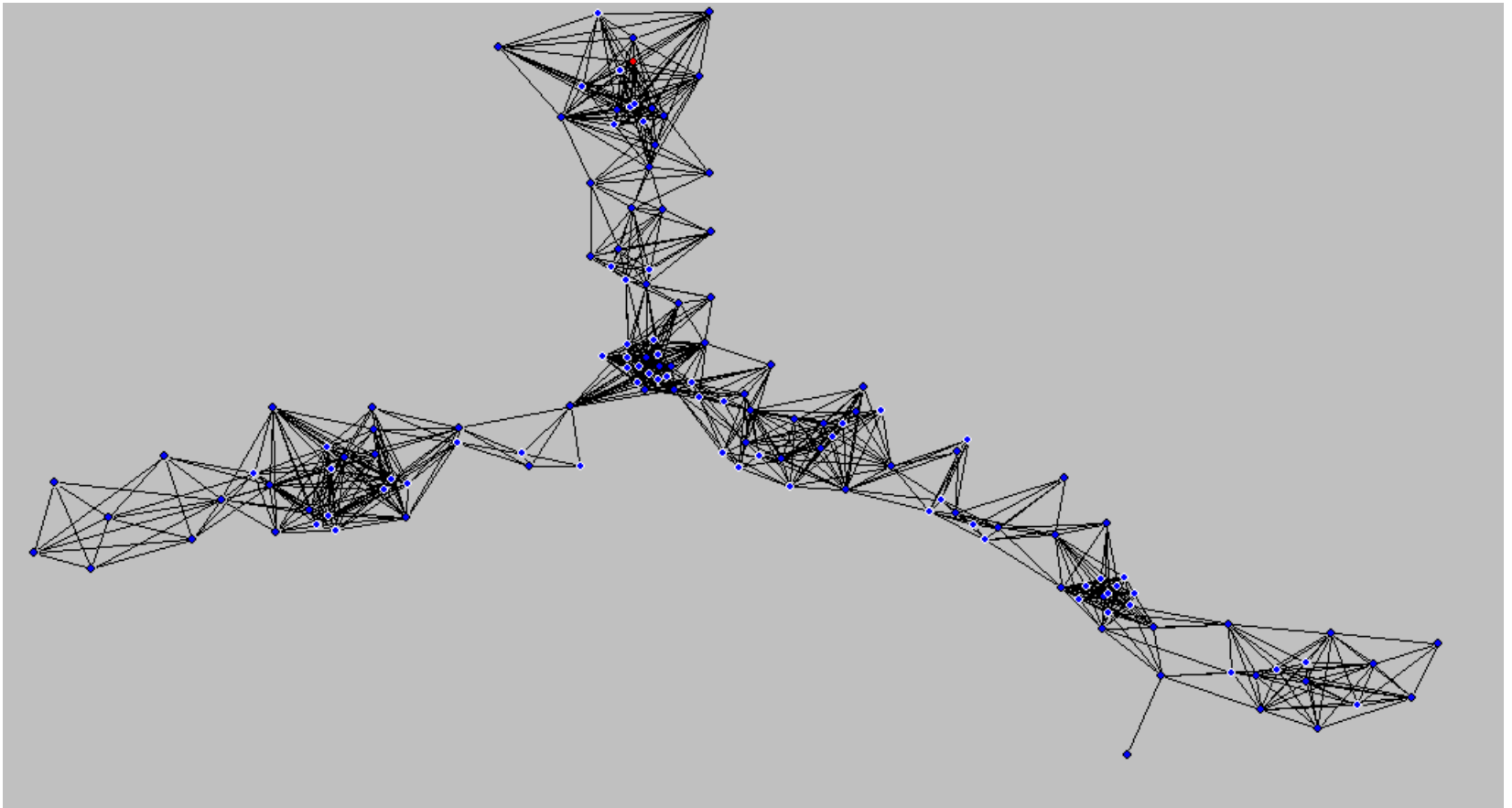
Identification of contig non-linearity

Example with Q-clone:



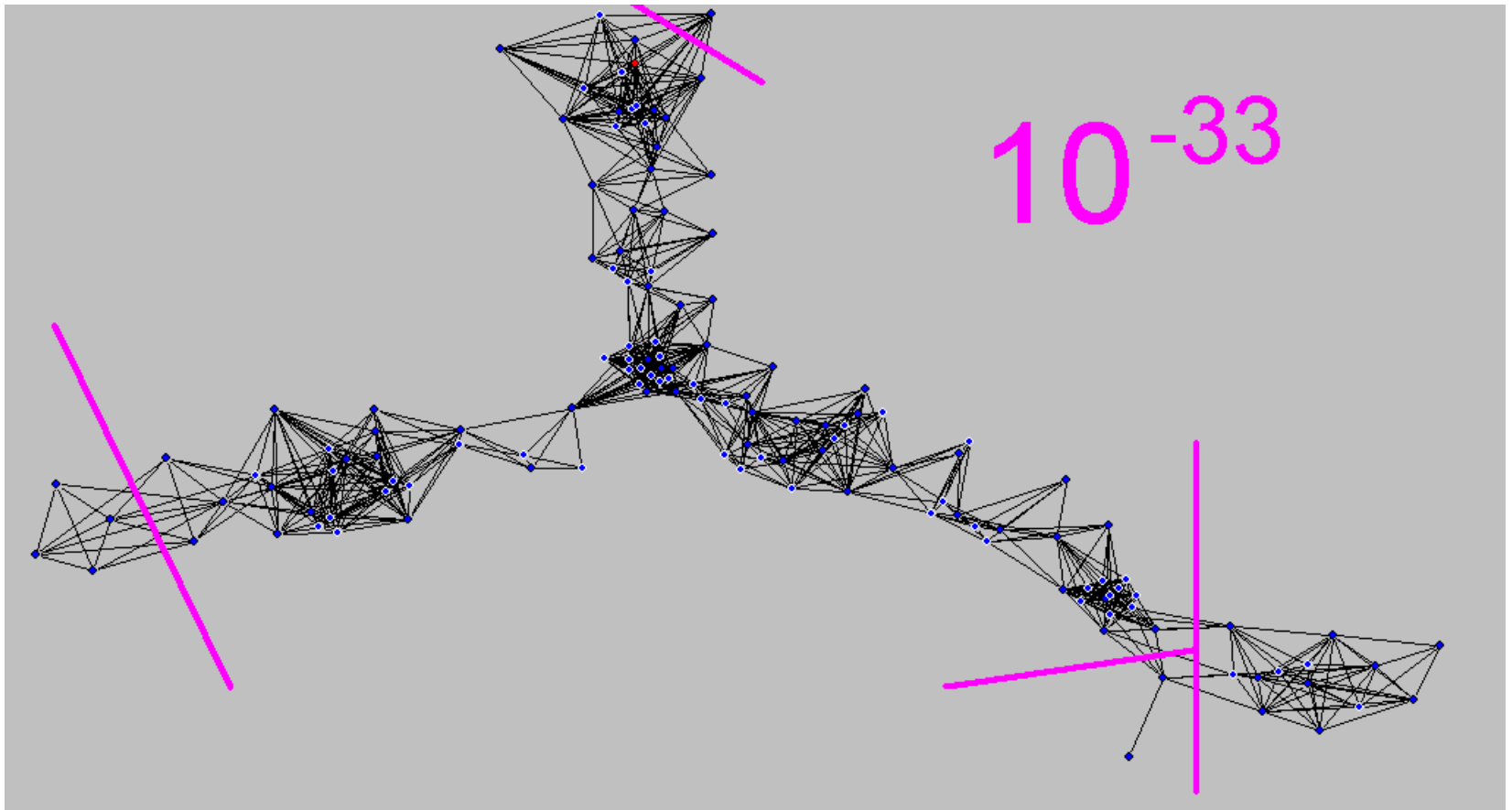
Identification of contig non-linearity

Increasing cutoff stringency alone may lead to dissolving rather than linearization of the contig:



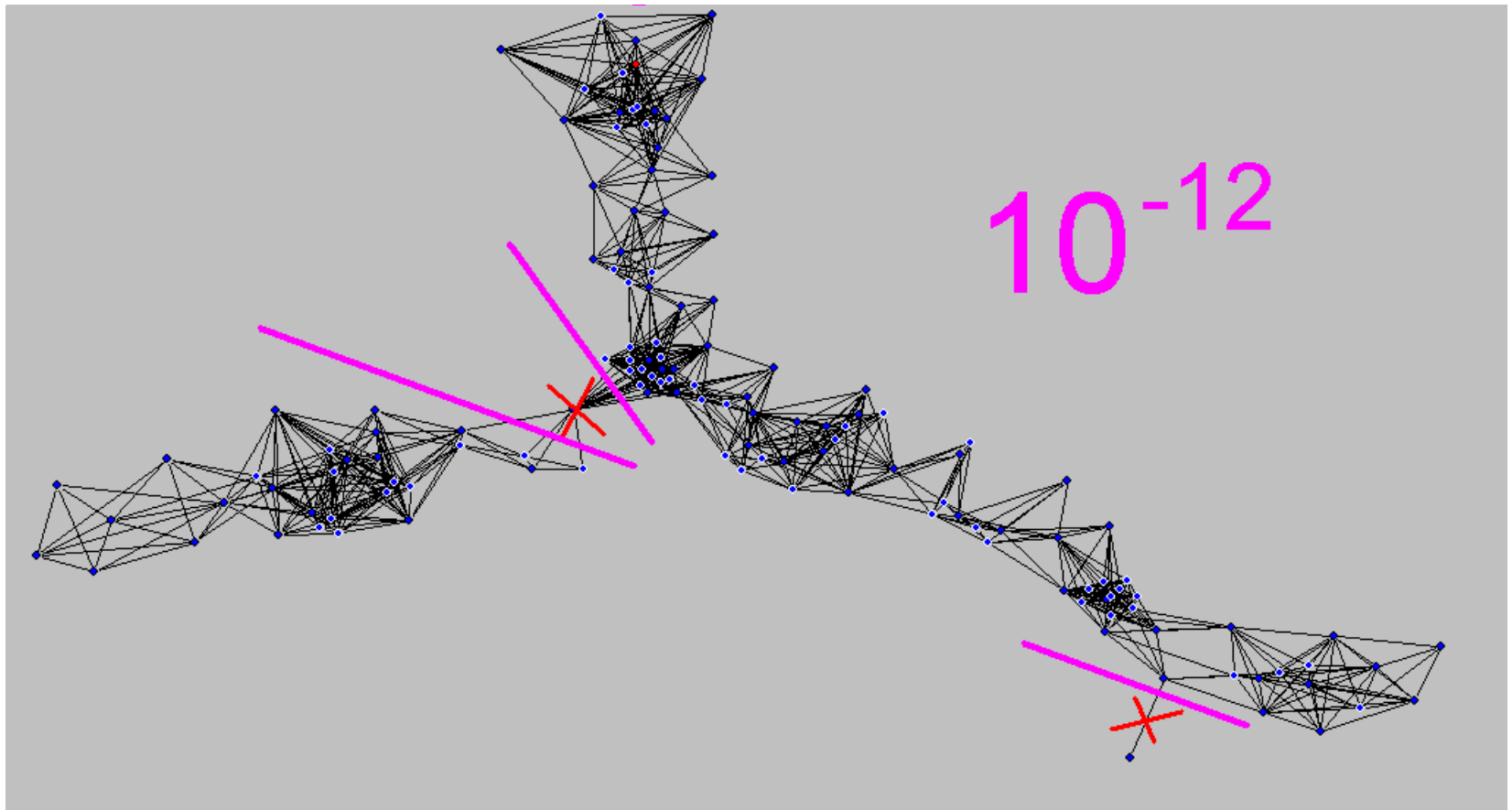
Identification of contig non-linearity

Increasing cutoff stringency alone may lead to dissolving rather than linearization of the contig:

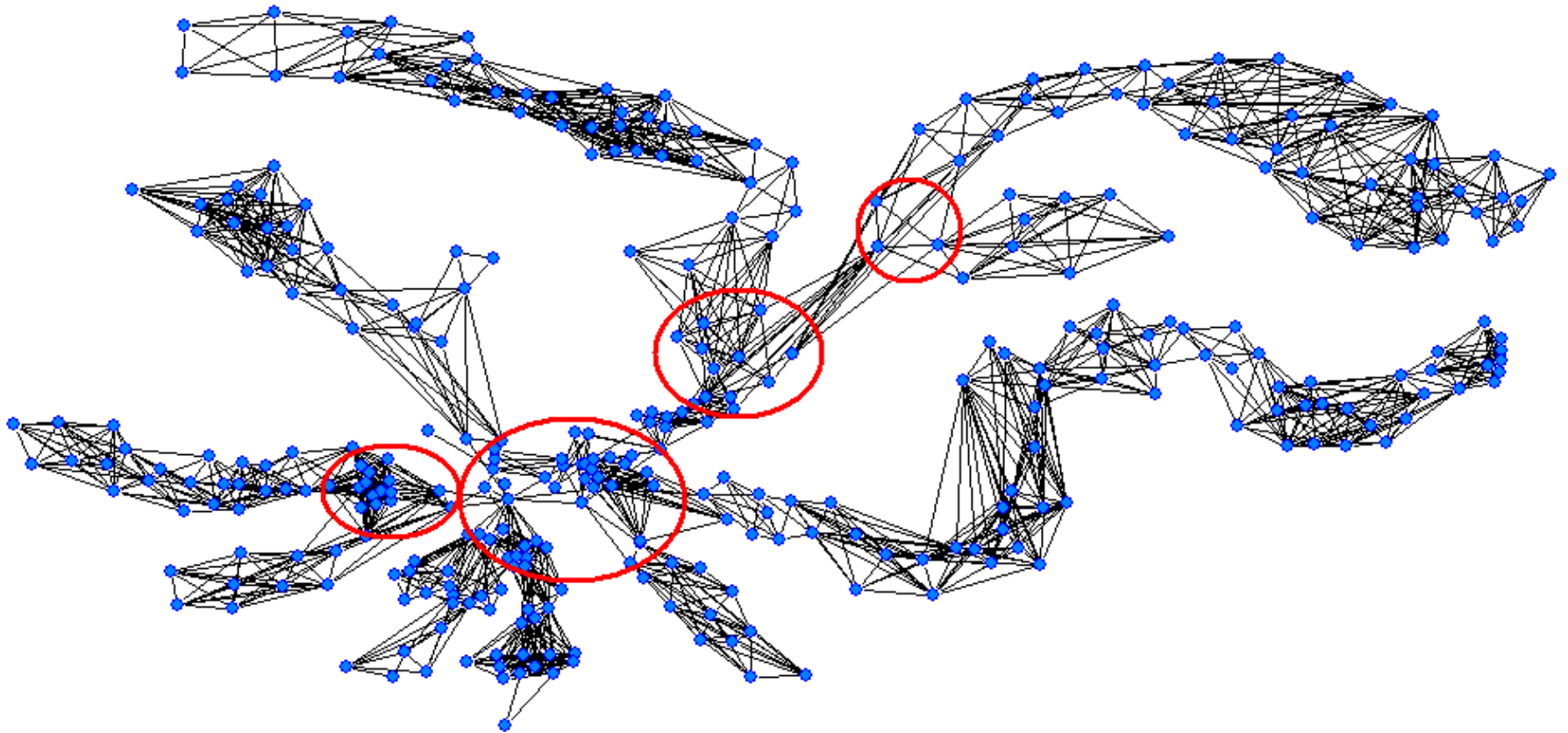


Identification of contig non-linearity

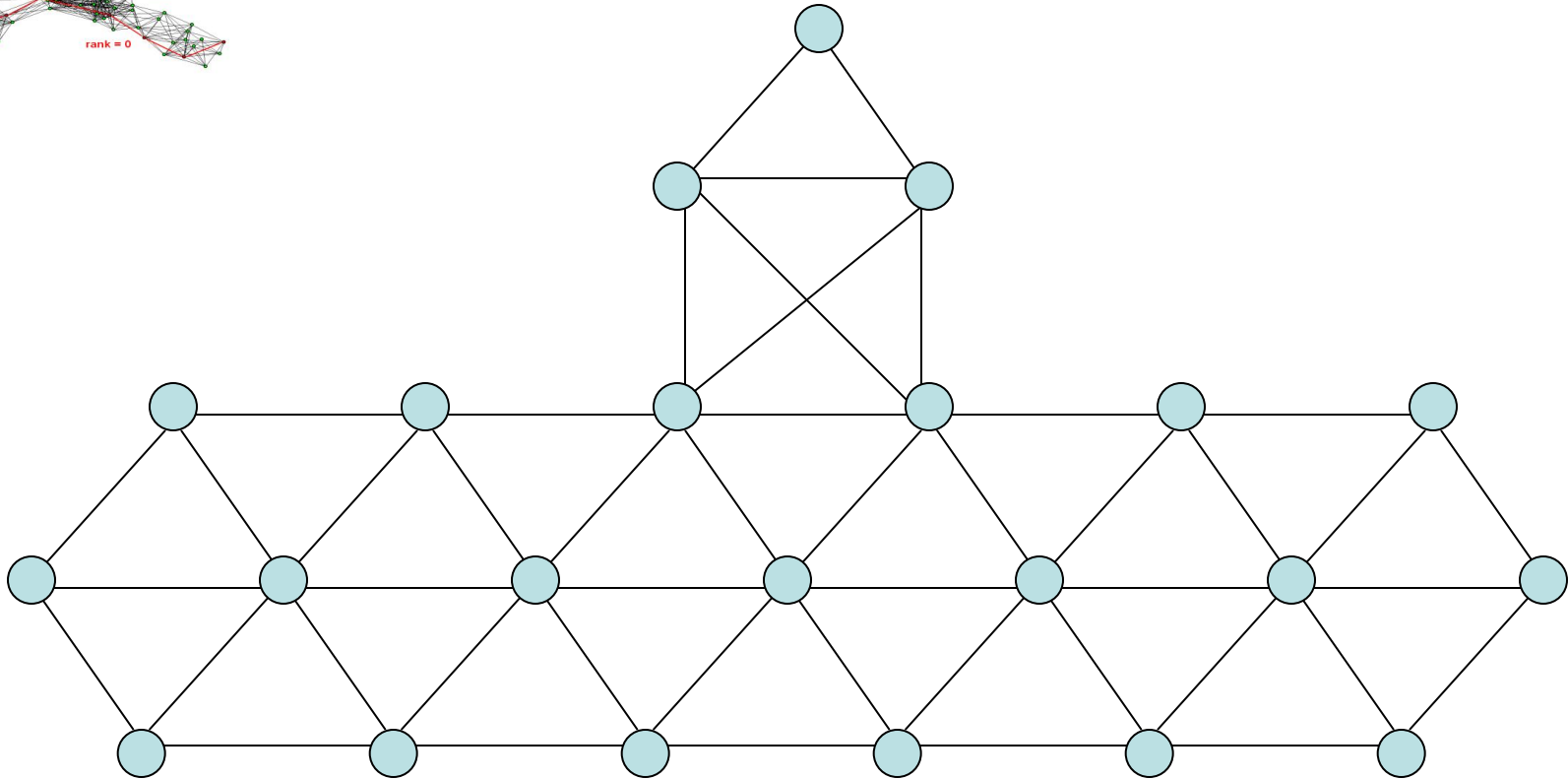
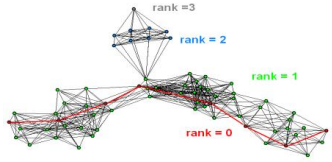
By excluding Q-clones we obtain linear clusters even at a liberal cutoff:



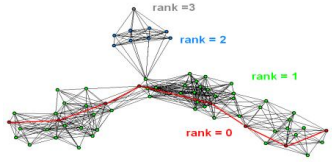
Detection of branching points



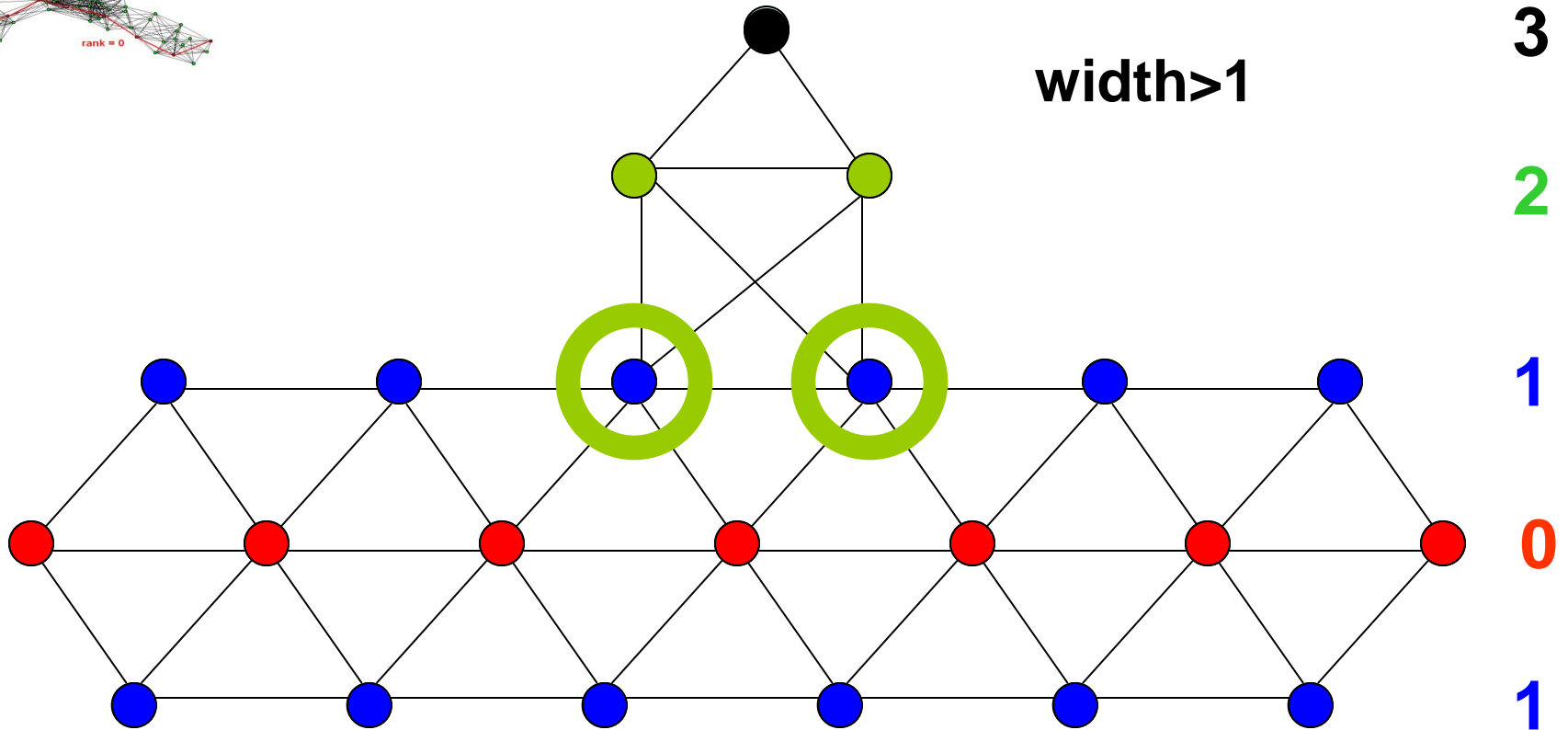
Detection of branching points



Detection of branching points



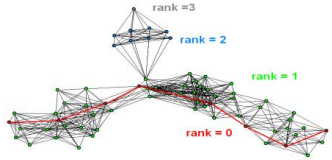
width>1



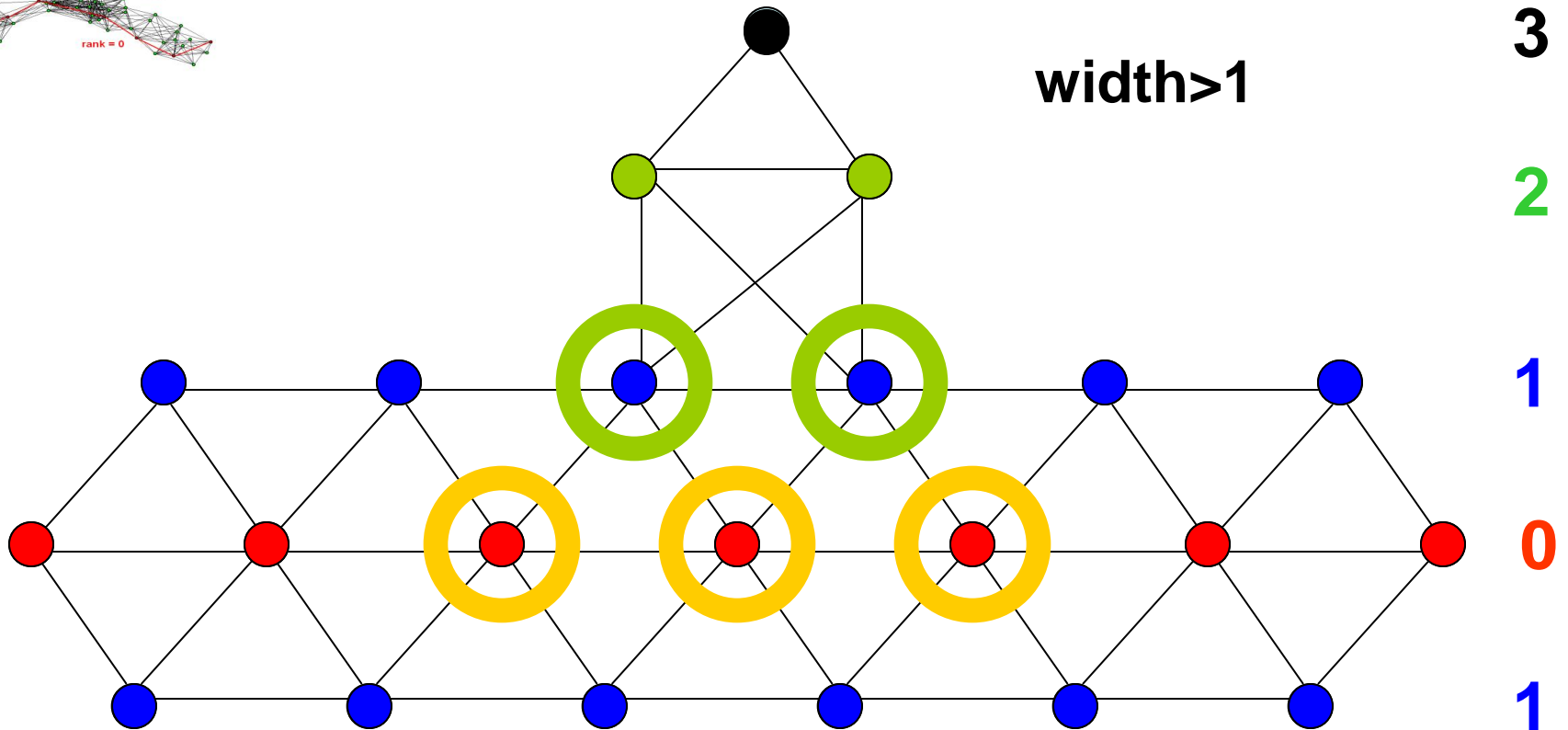
ranks

Identification of clones of rank 1 overlapped with clones of rank 2

Detection of branching points



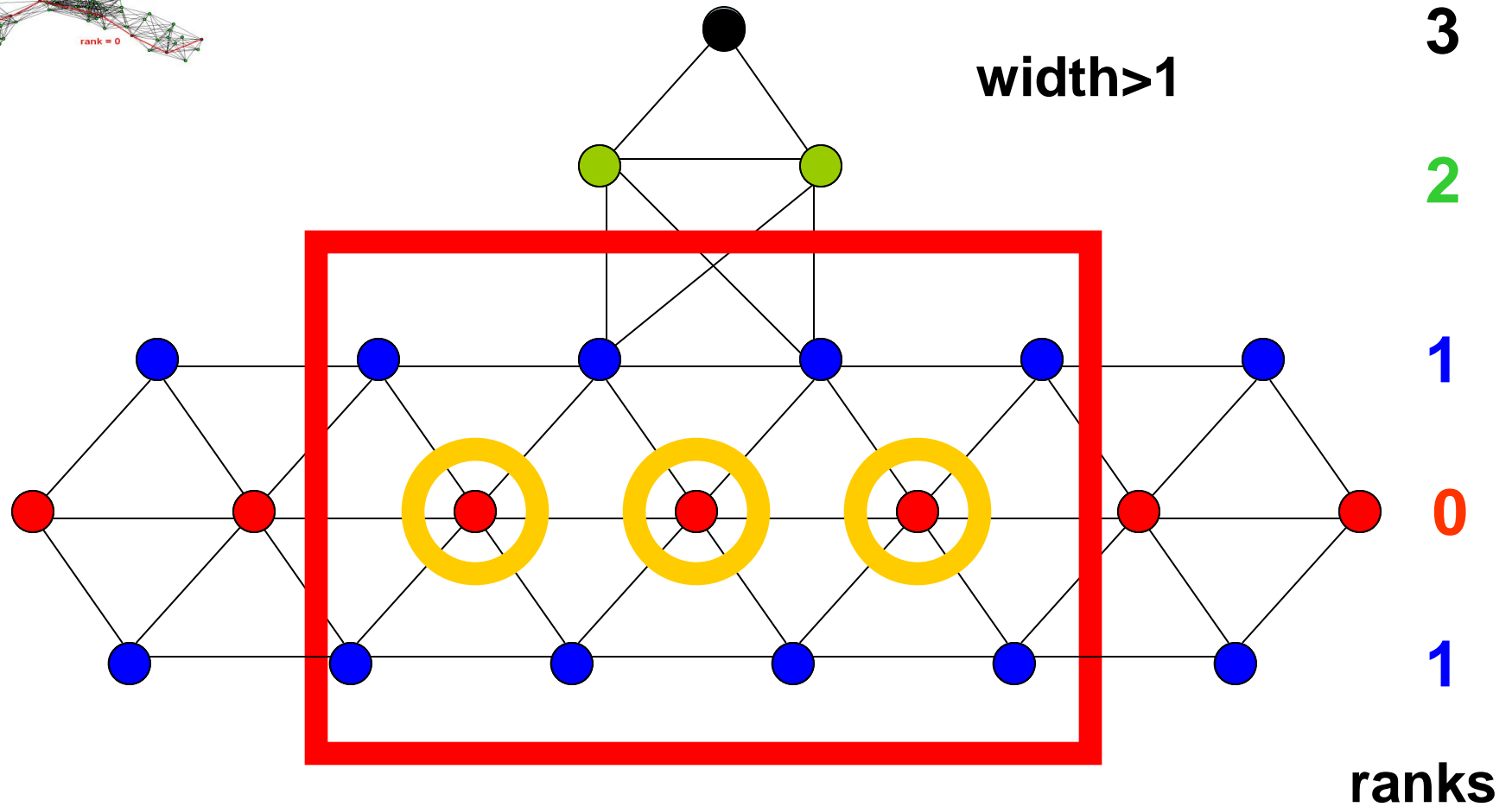
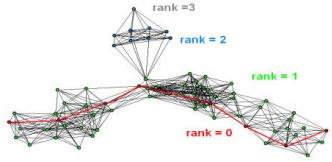
width>1



Identification of clones from diametric path overlapped with the detected clones

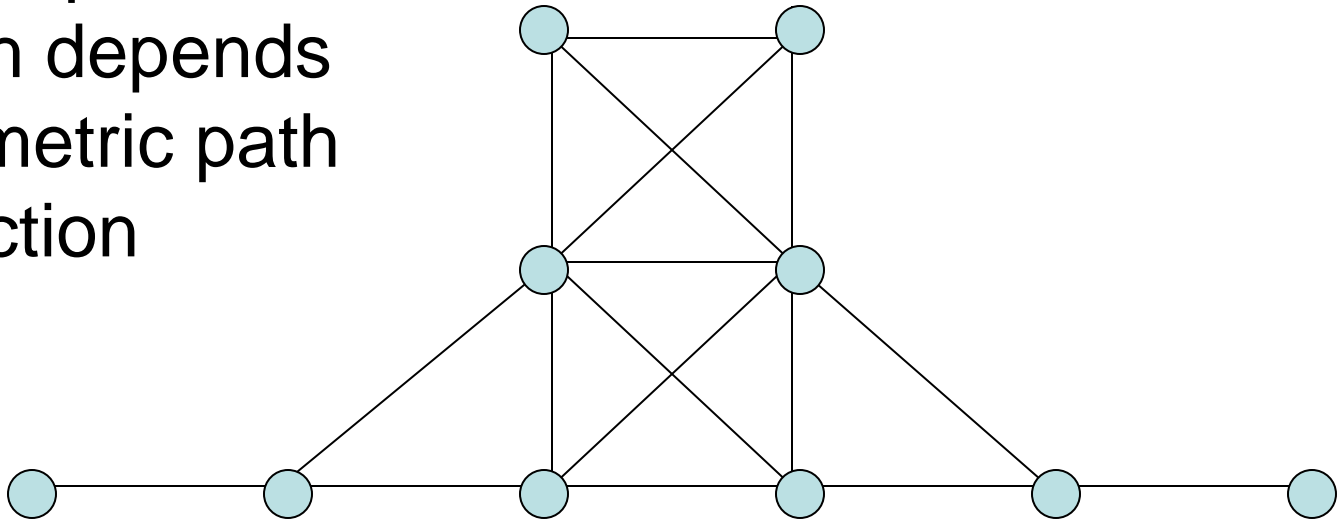
ranks

Detection of branching points



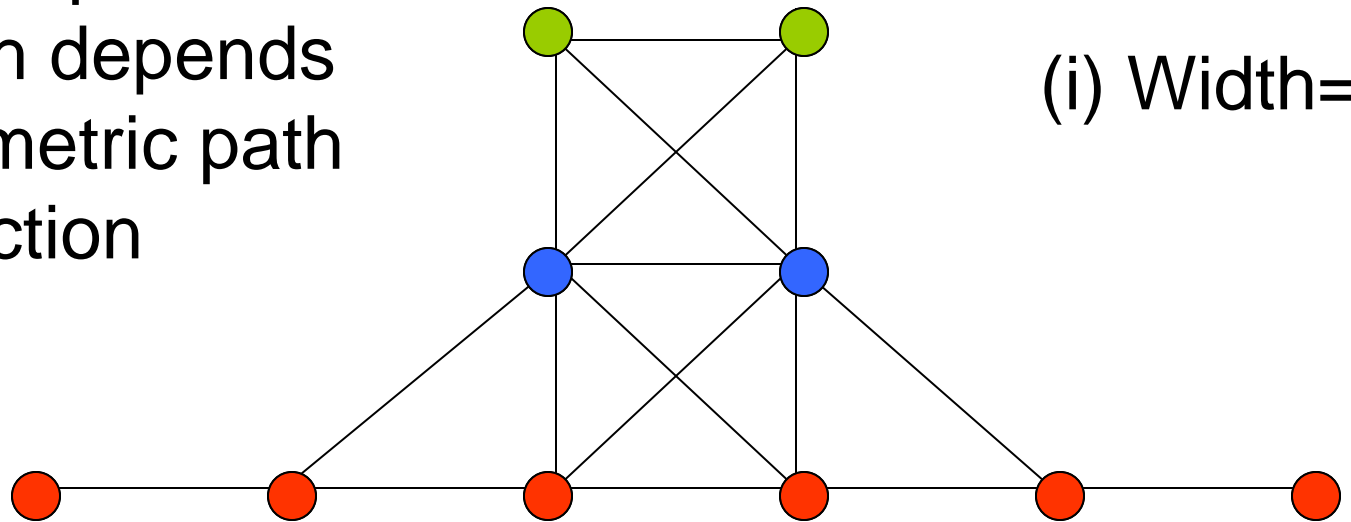
Detection of branching points

A more complicated case: width depends on the diametric path selection



Detection of branching points

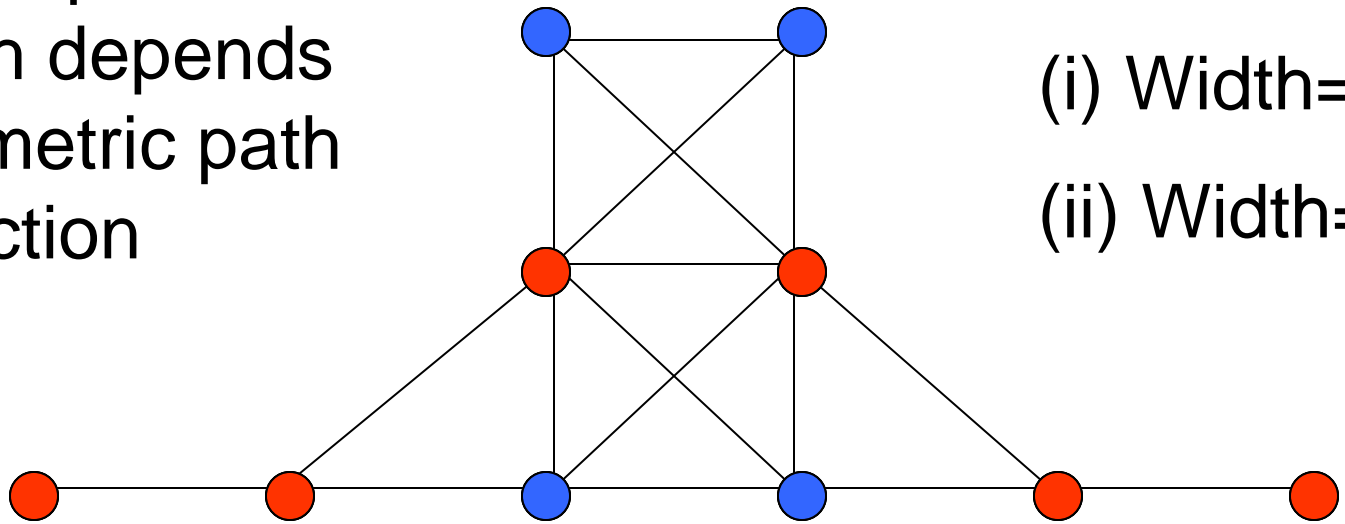
A more complicated case: width depends on the diametric path selection



(i) Width=2

Detection of branching points

A more complicated case: width depends on the diametric path selection



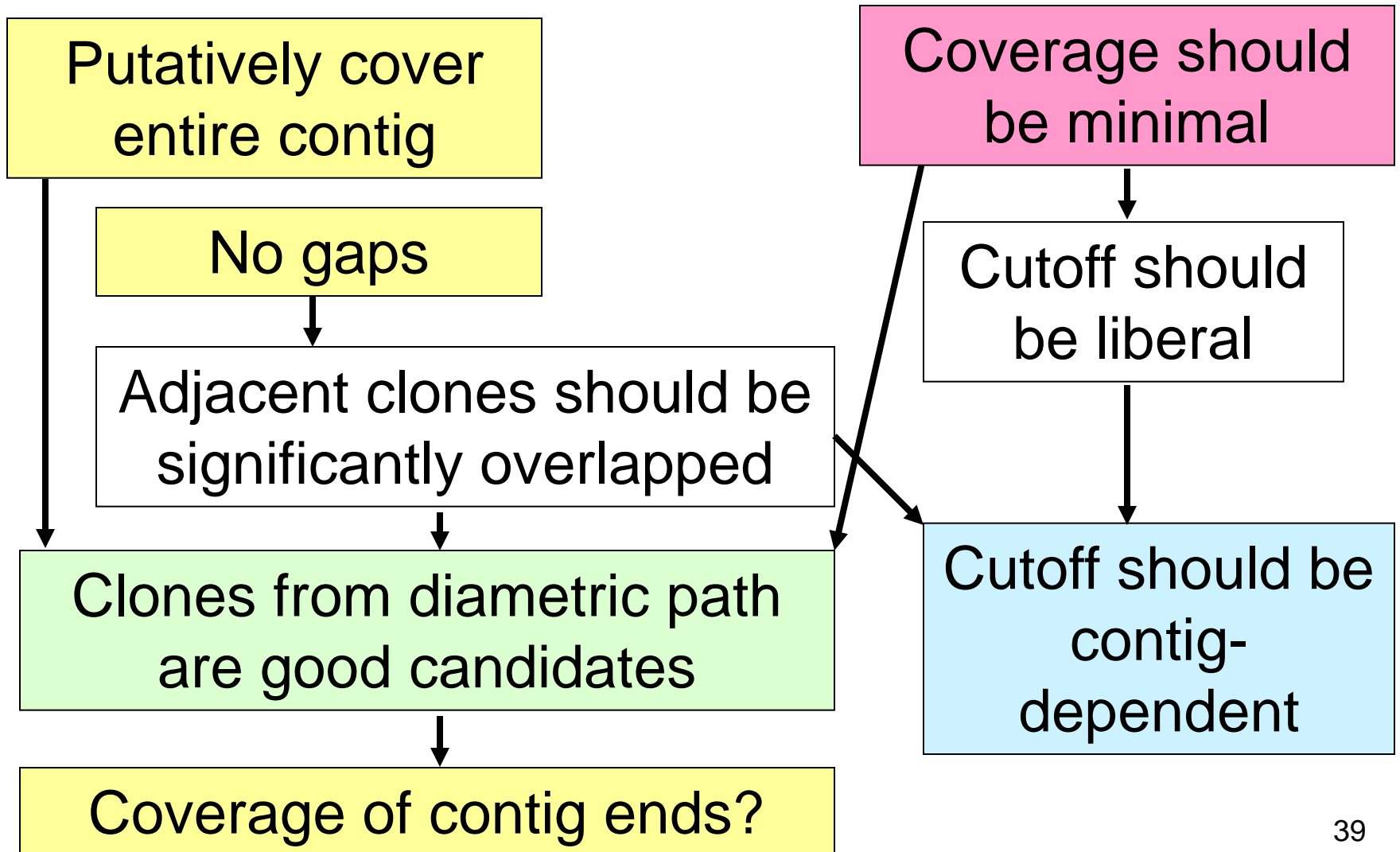
(i) Width=2

(ii) Width=1

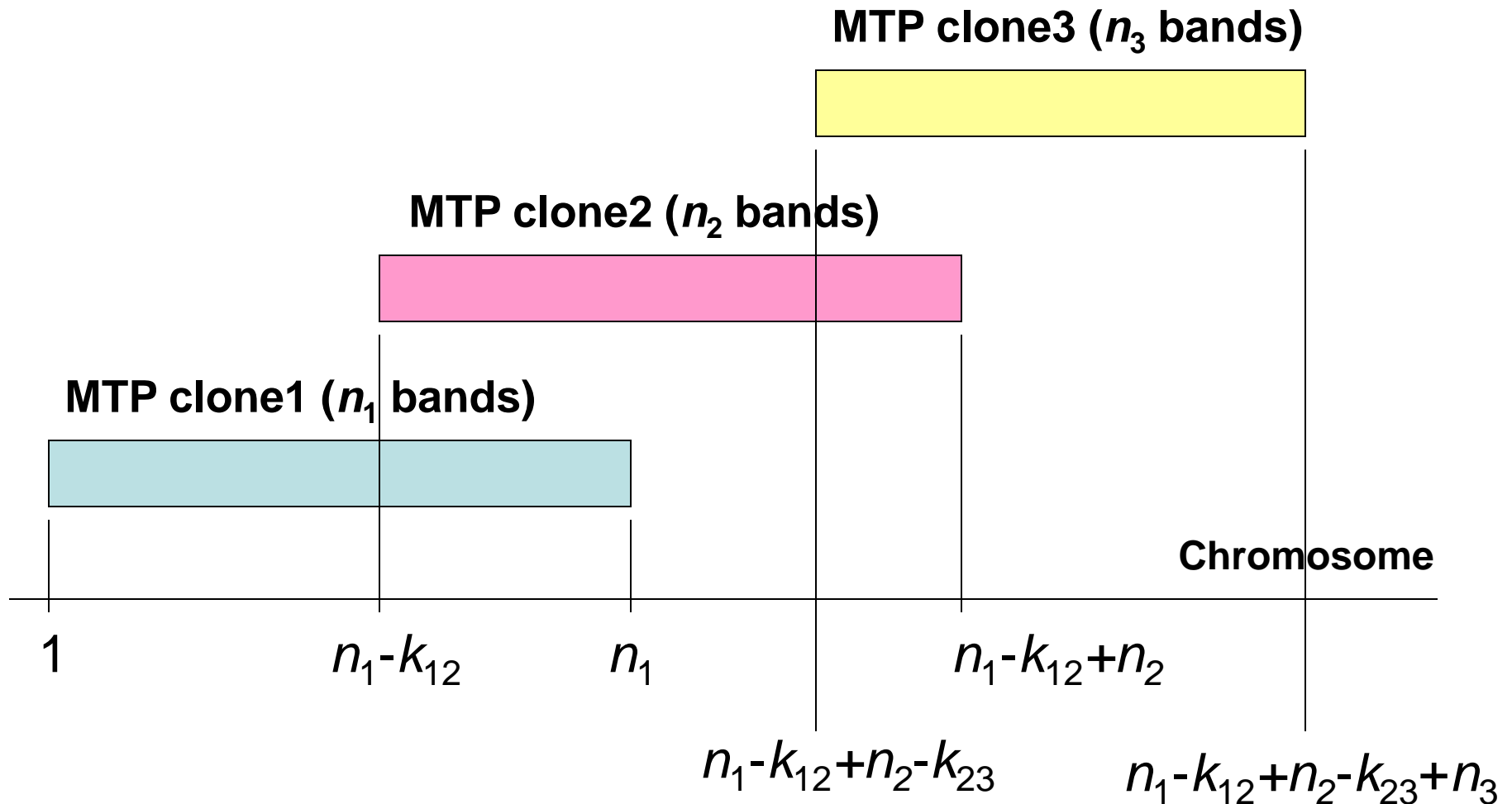
Warning:

- Low quality of fingerprinting?
- False clone overlaps?
- Chimerical clones?

MTP-selection



Clone-end coordinates



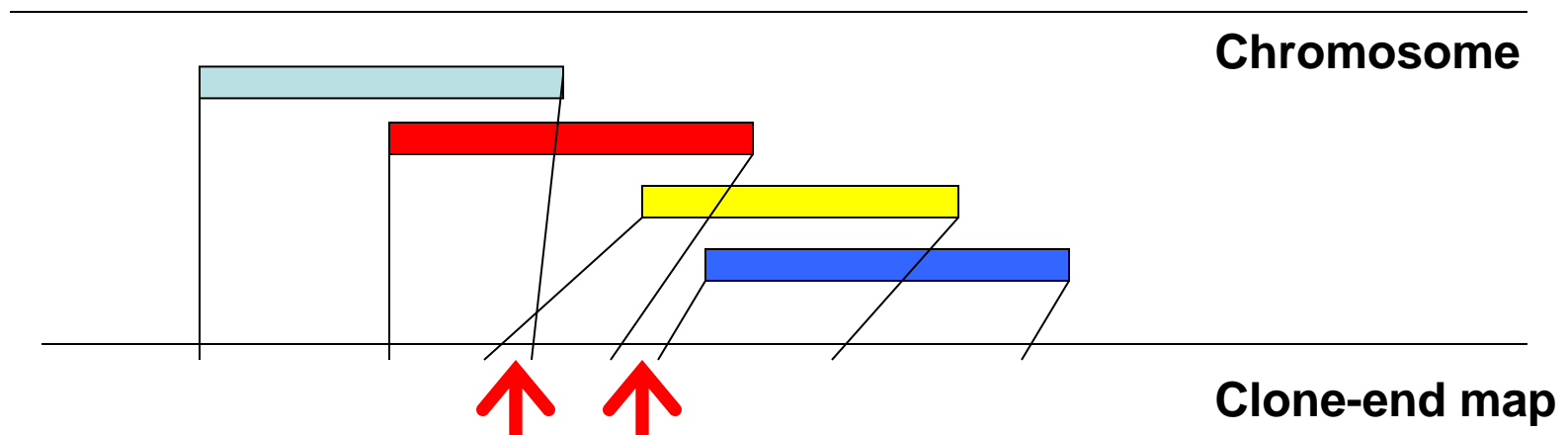
Clone-end coordinates

Problems:

- Missing bands in clones
- False bands
- Bands with the same size (e.g., repeats)



Real overlap $\leftarrow ? \rightarrow$ Coordinate-based overlap

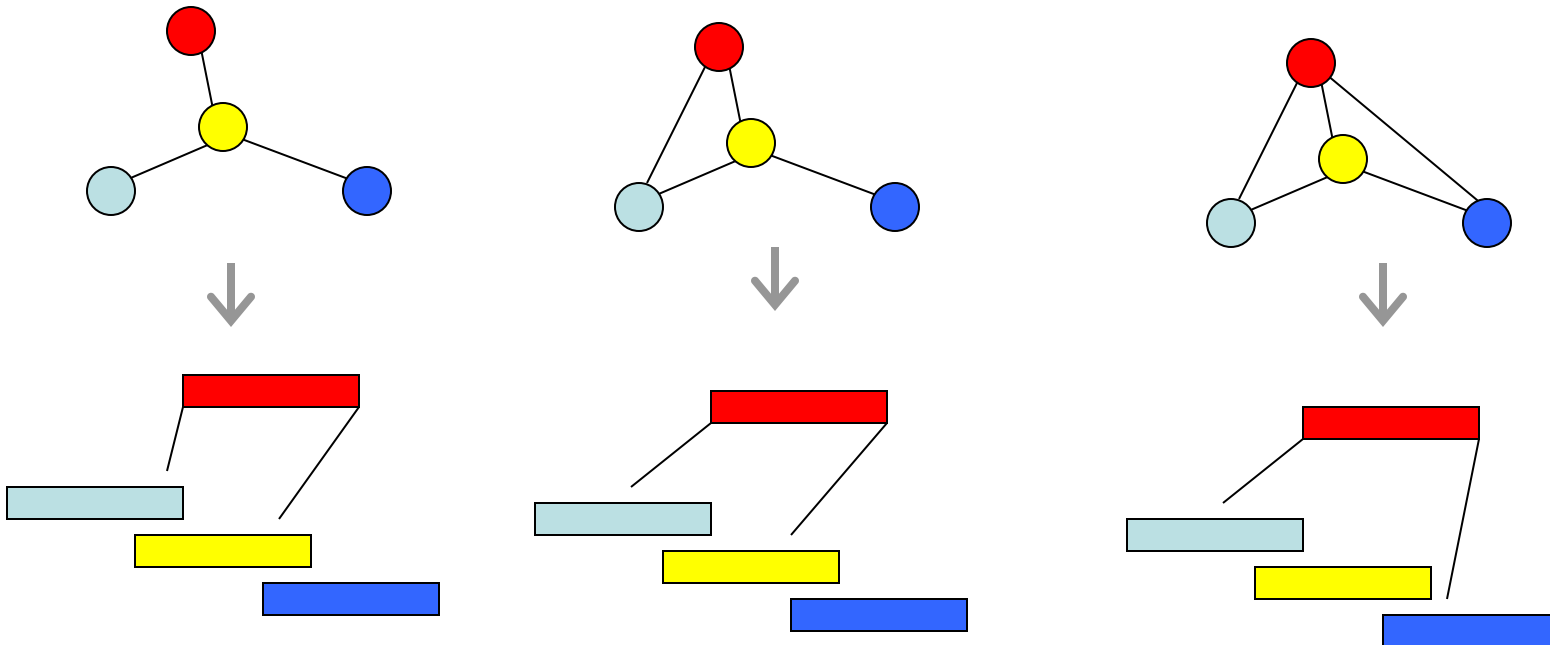


Clone-end coordinates

(for clones not from MTP)

Each such clone overlaps with 1 to 3 MTP clones.
In defining the coordinates we try to:

- Reduce the rate of contradictions
- Not adding clones to the ends of the MTP



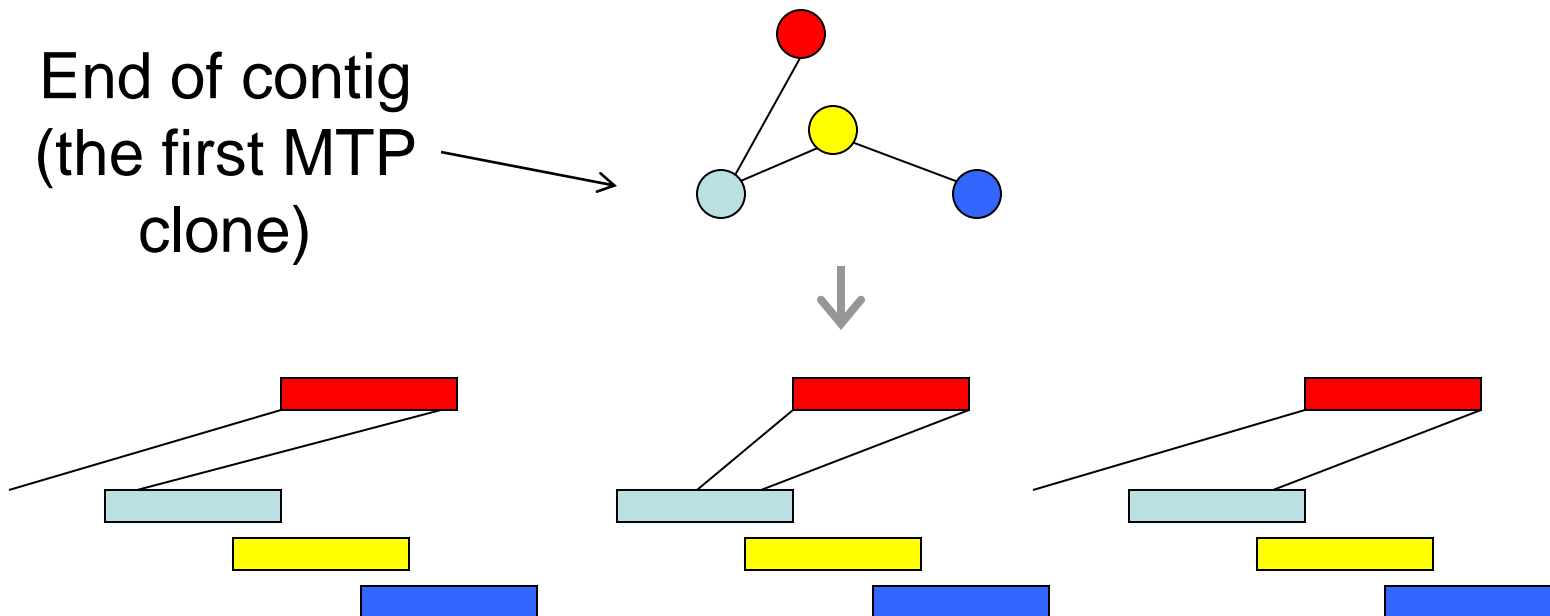
Clone-end coordinates

(for clones not from MTP)

Left-end coordinate ≤ 0

→ add to MTP or substitute the first MTP clone

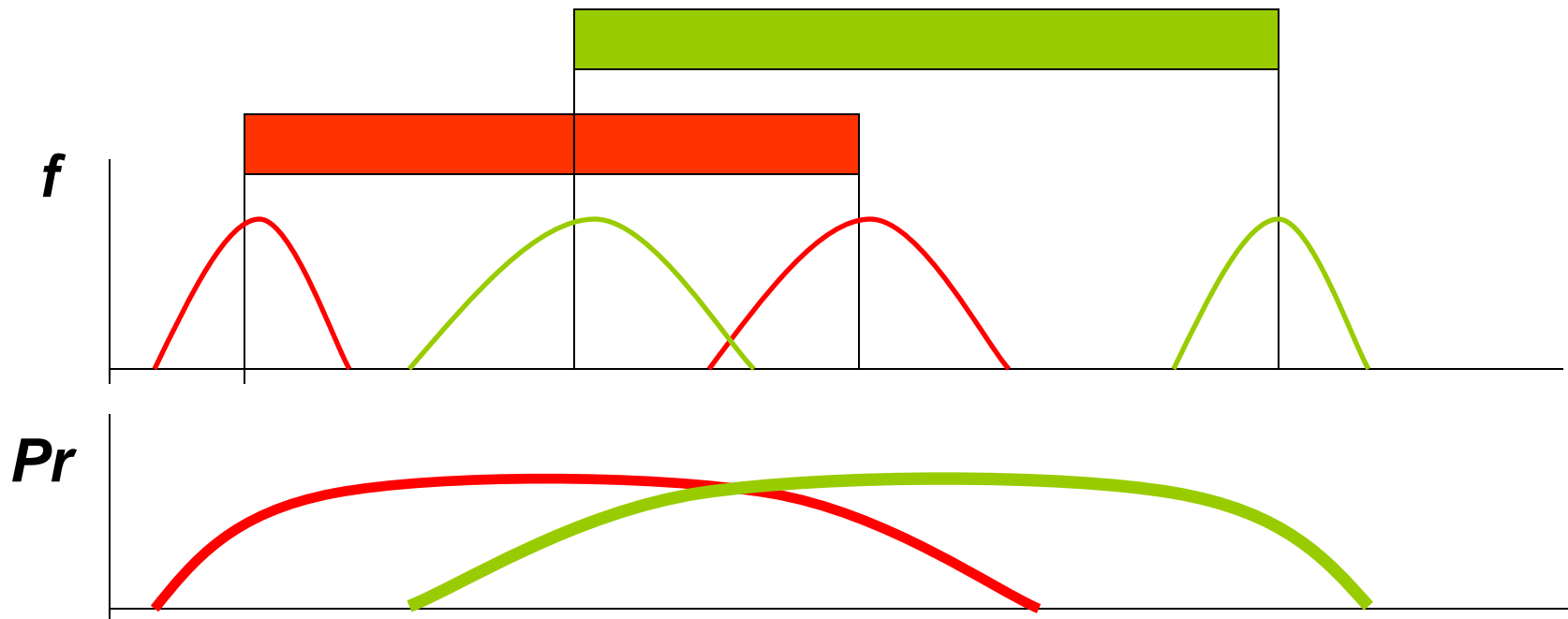
- Reduce the rate of contradictions
- Not adding clones to the ends of the MTP



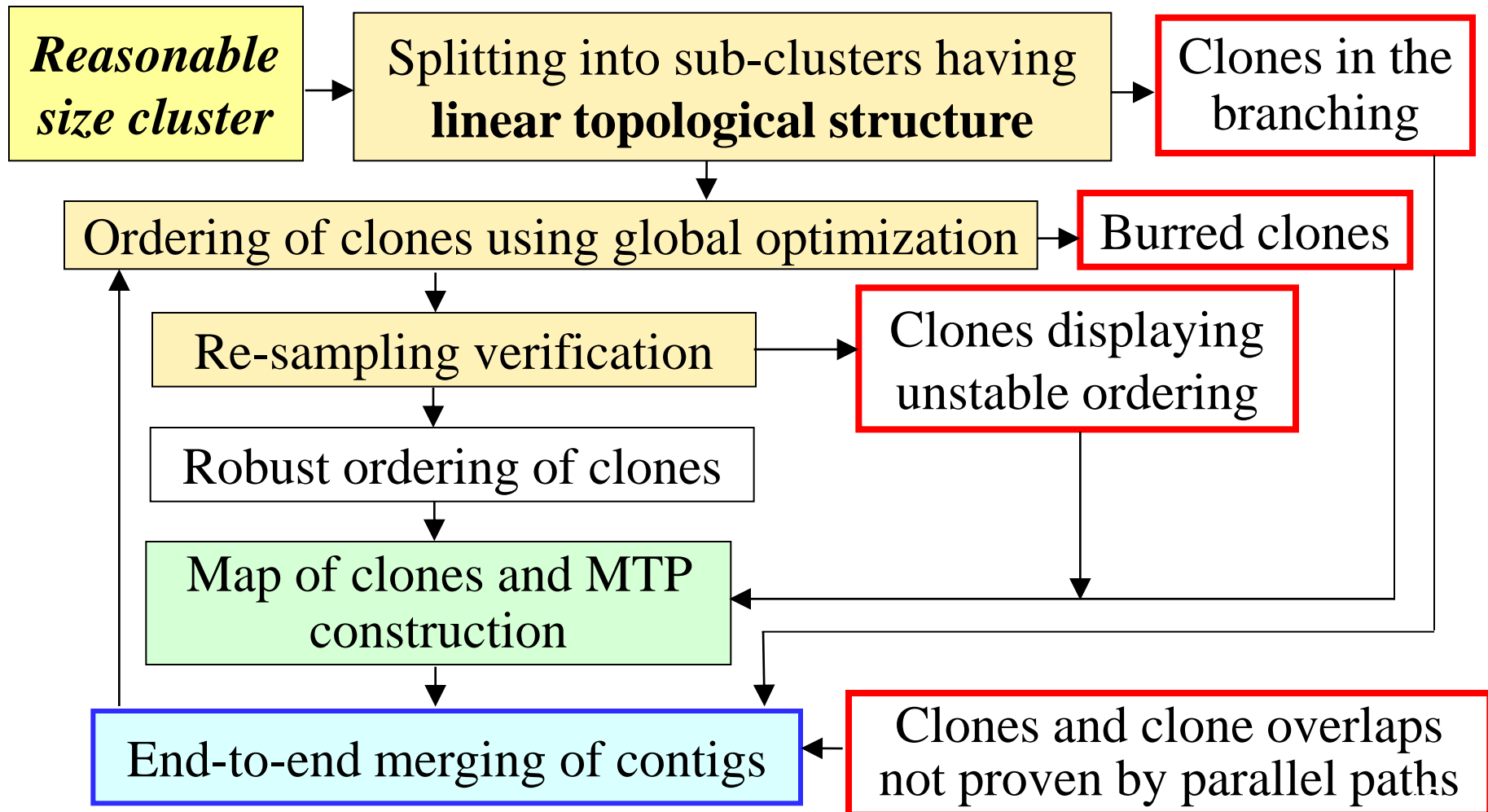
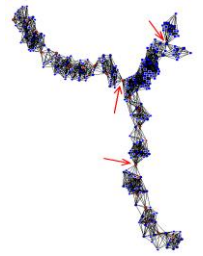
Clone-end coordinates

Taking into account the uncertainty of the estimated coordinates of the clone ends

- Resampling of bands (jackknife)
- Distribution (interval) rather than exact (point) estimation



Adaptive contig assembly: Clustering coordinated with ordering & verification



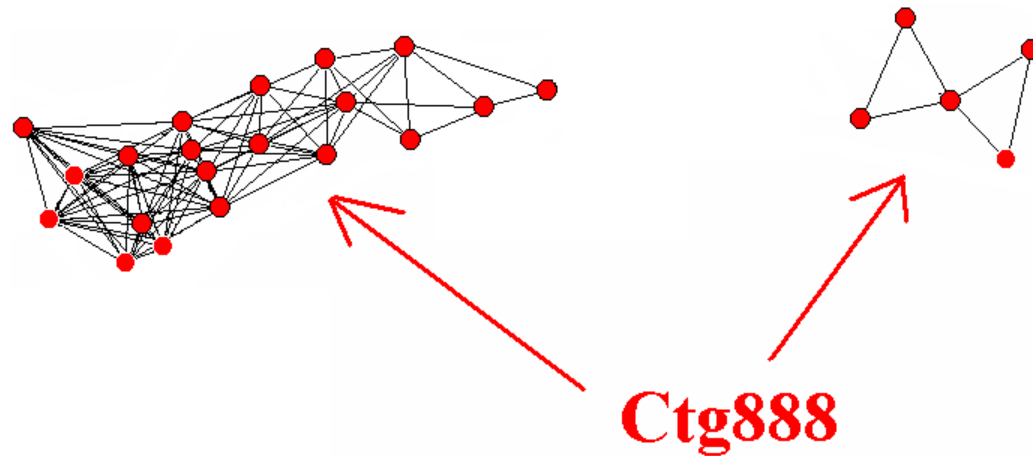
Open questions

- More accurate calculation of p -values
- Taking into account band frequencies
- Band frequency → band “haplotypes”
- Coordinates of clones
- More effective identification of Q-clones and Q-overlaps (e.g., using info on bands)
- Automatic linearization of difficult places

LTC for FPC users

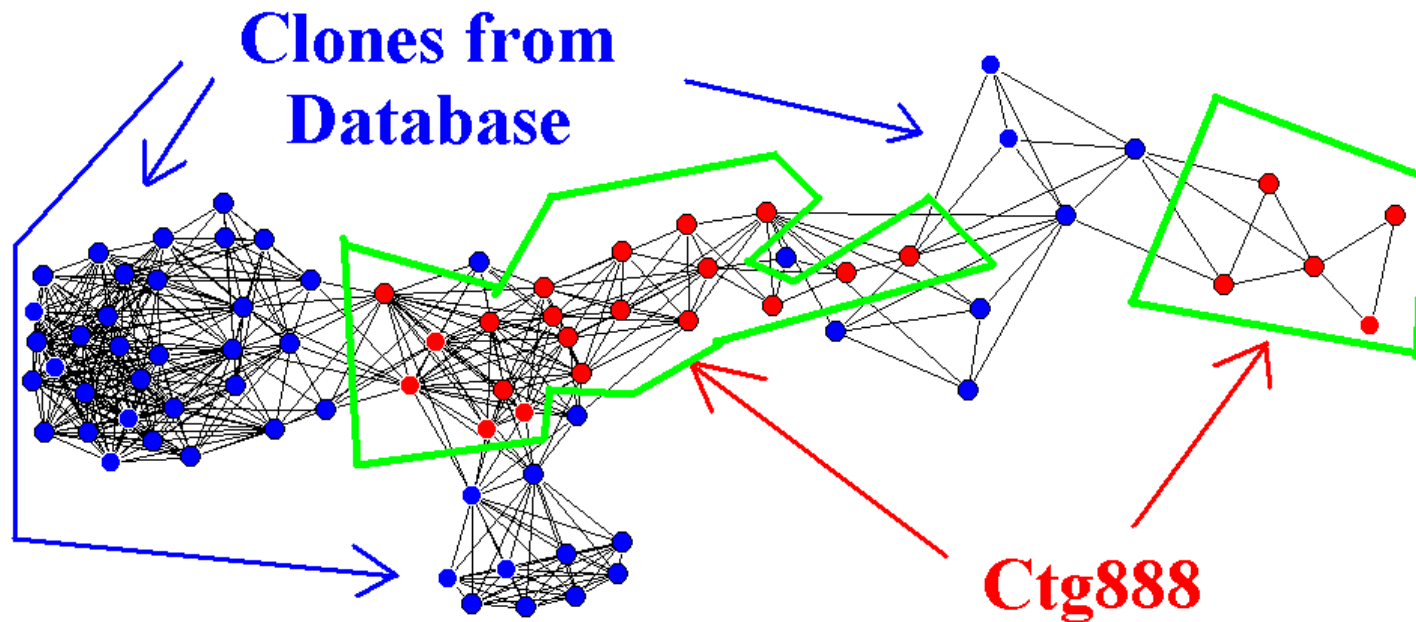
Testing FPC contigs by using LTC

Some FPC contigs consist of **non-connected** parts:



Testing FPC contigs by using LTC

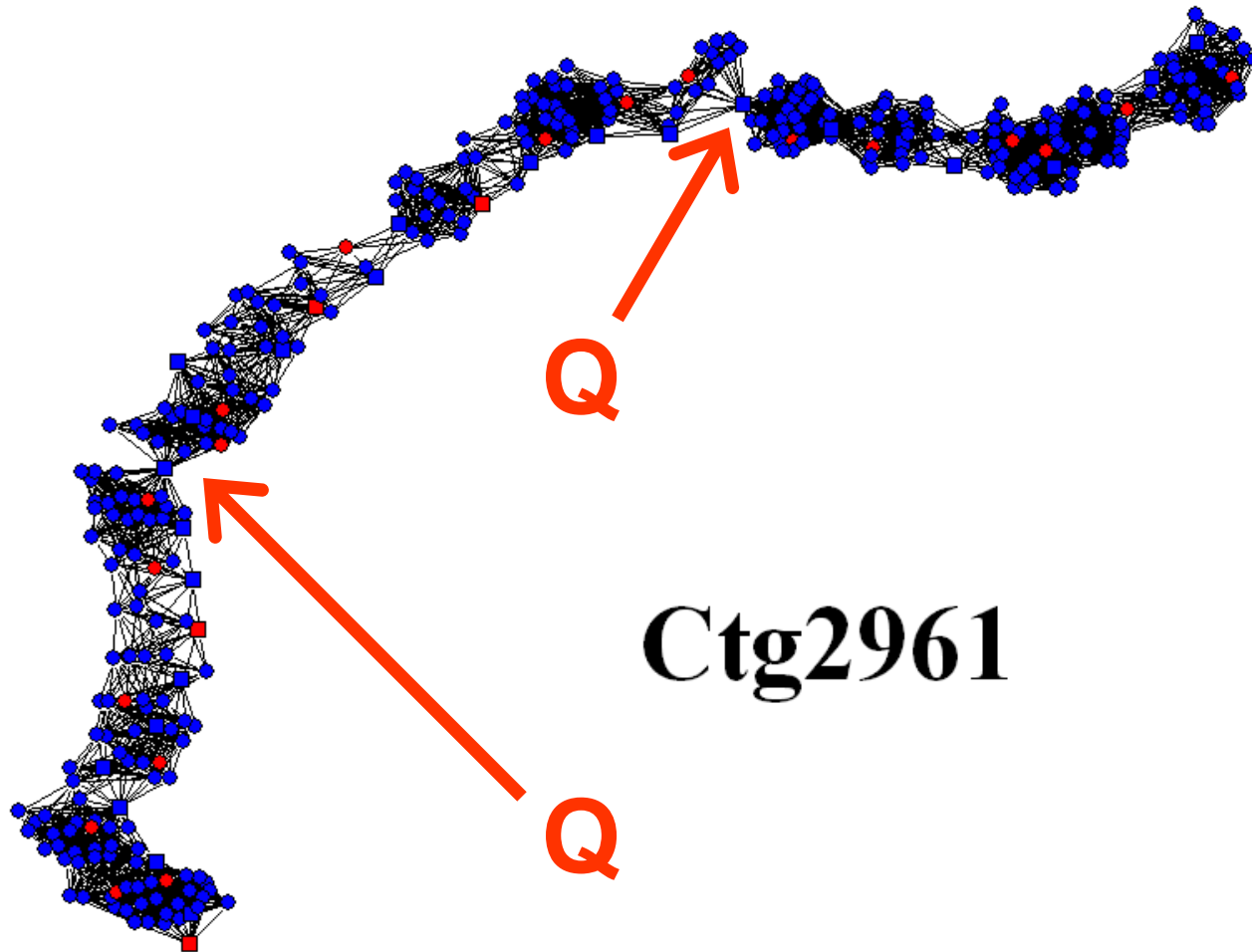
Some FPC contigs consist of **non-connected** parts:



Gap repaired by adding clones

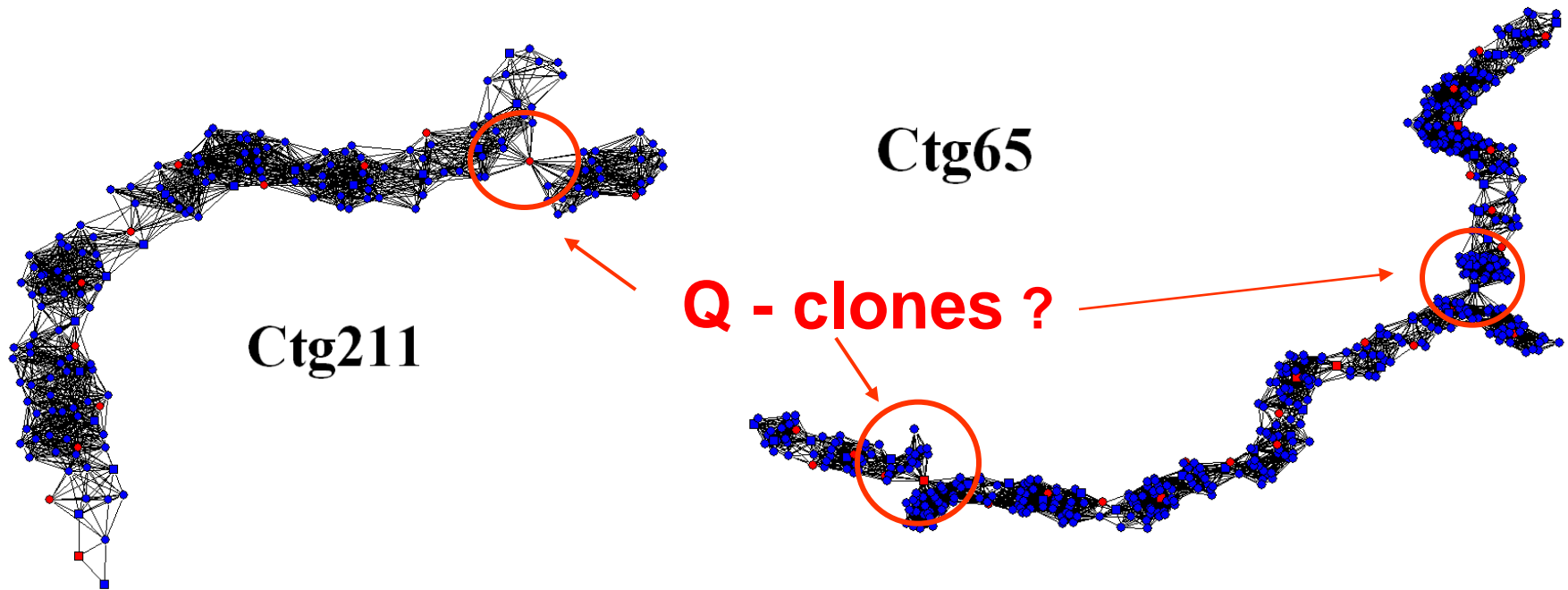
Testing FPC contigs by using LTC

Wheat 1B: In some FPC contigs internal connections are via Q-clones (chimerical contigs?)



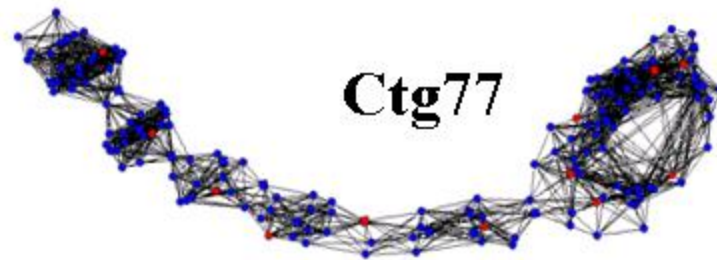
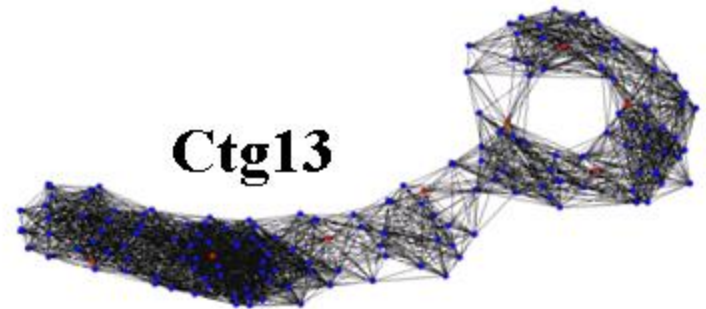
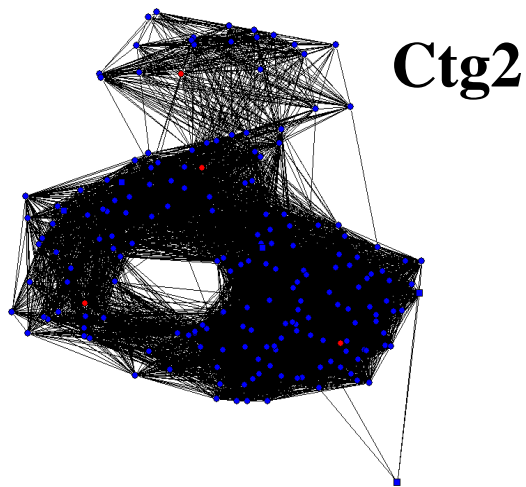
Testing FPC contigs by using LTC

Wheat 1B: Q-clones may cause non-linearity:



Testing FPC contigs by using LTC

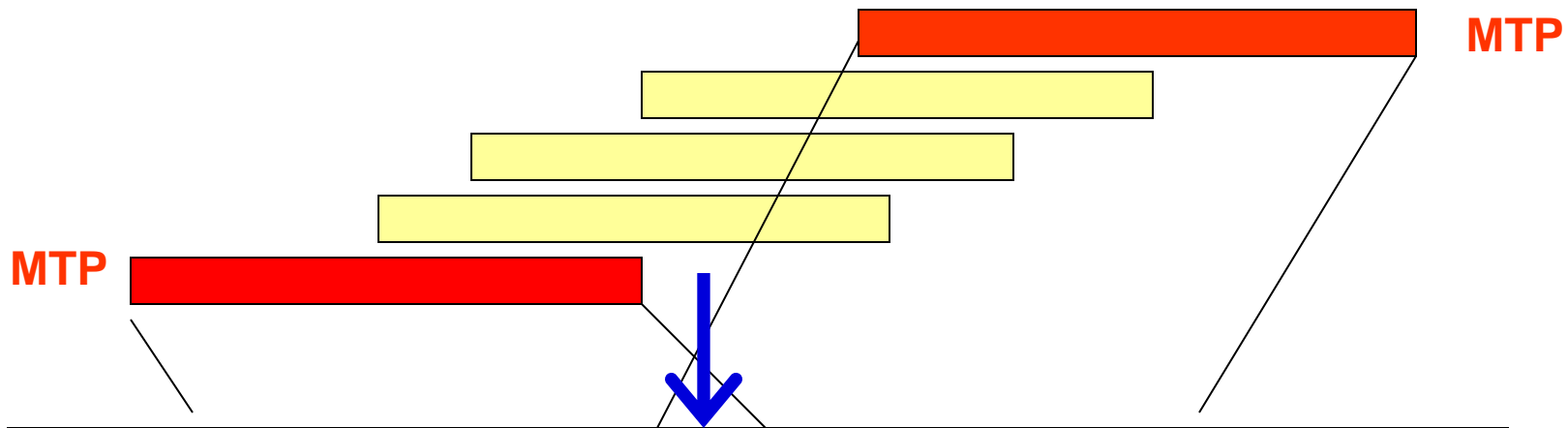
Examples of FPC contigs with non-linear topology, and “cycles”, **without Q-clones**



Edges represent significant overlaps (with cutoff **1e-25** Sulston score). Increasing the stringency up to **1e-75** does **not help here to get a non-trivial linearization!**

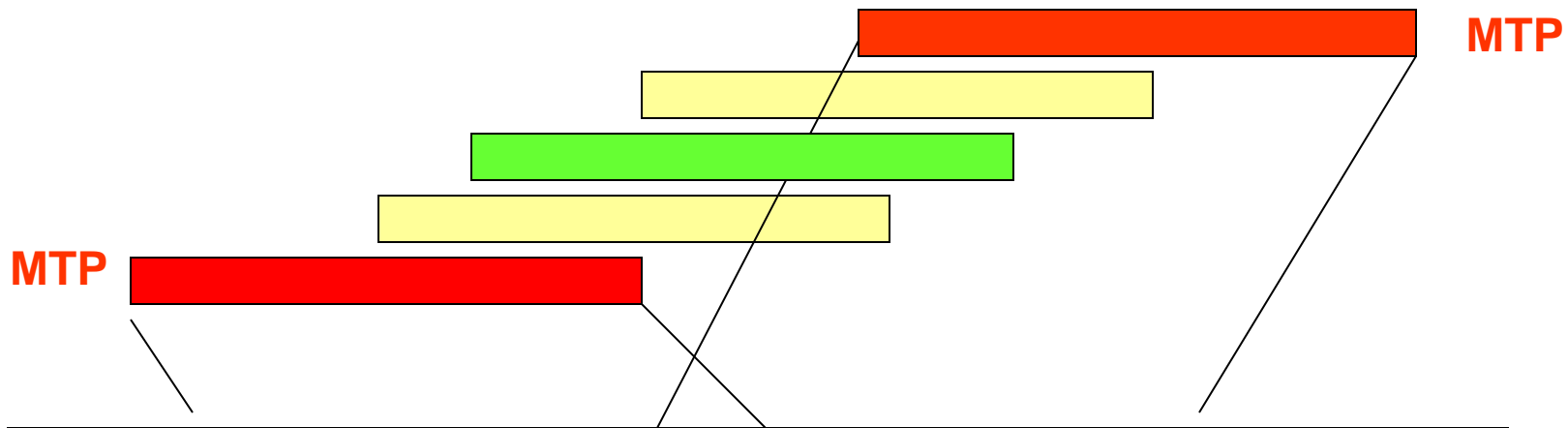
Testing FPC contigs by using LTC

About **30-60%** pairs of adjacent MTP clones in FPC contigs have no significant overlaps. This is caused by too liberal condition on overlap in MTP.
→ Putative sources for gaps



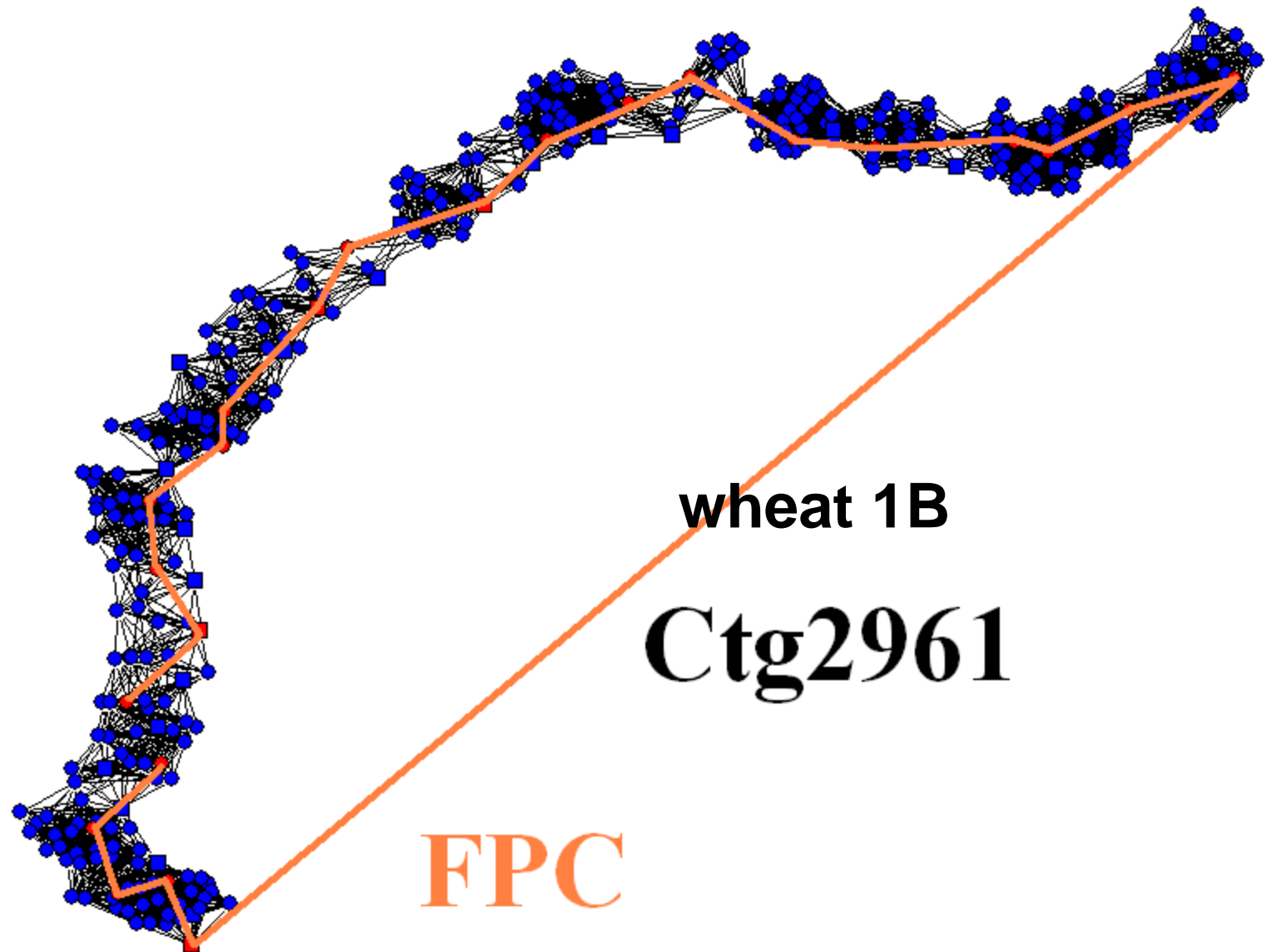
Testing FPC contigs by using LTC

About **30-60%** pairs of adjacent MTP clones in FPC contigs have no significant overlaps. This is caused by too liberal condition on overlap in MTP.
→ Putative sources for gaps

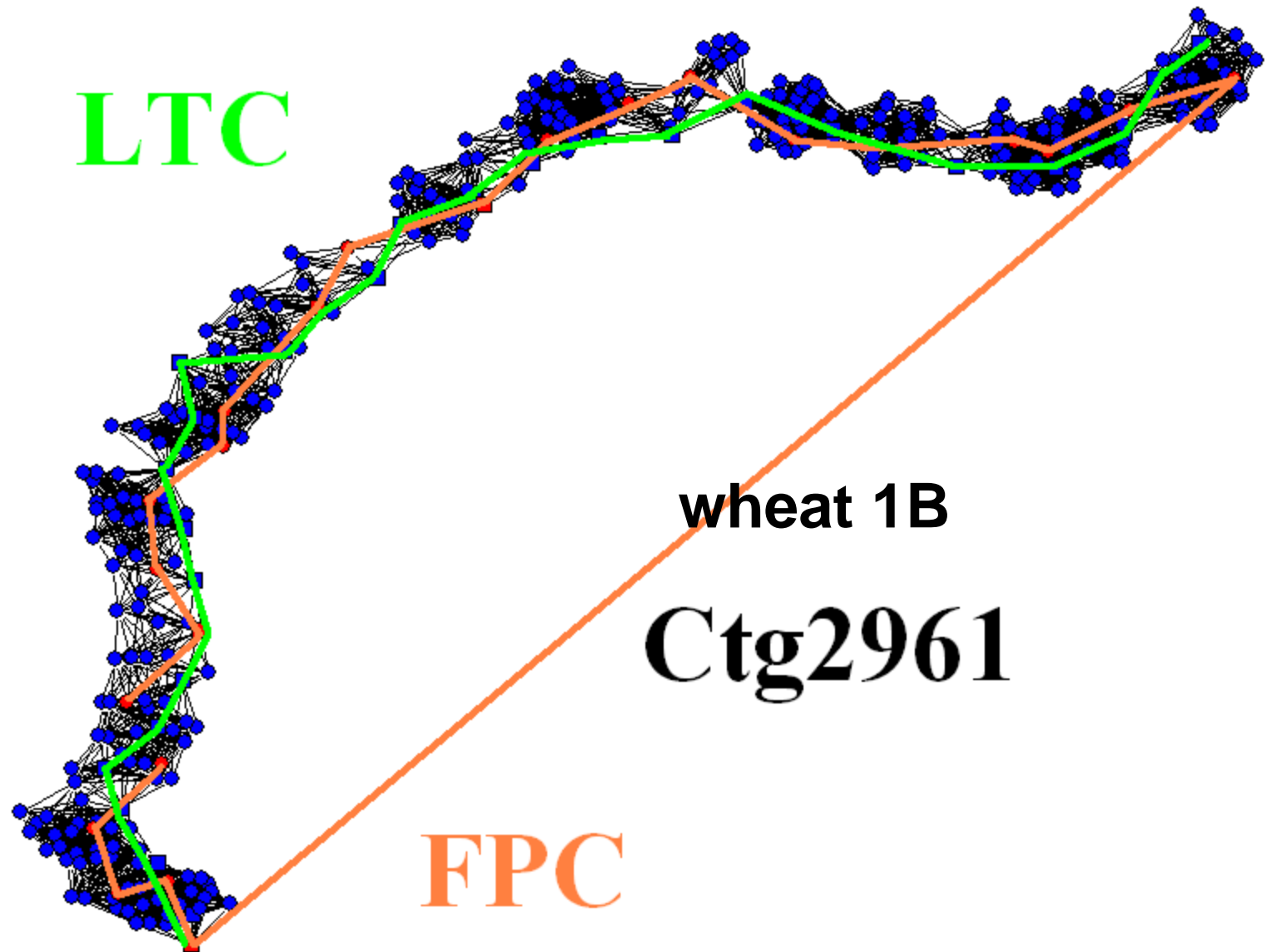


With correct ordering → gaps can be closed by complementing MTP with additional clones

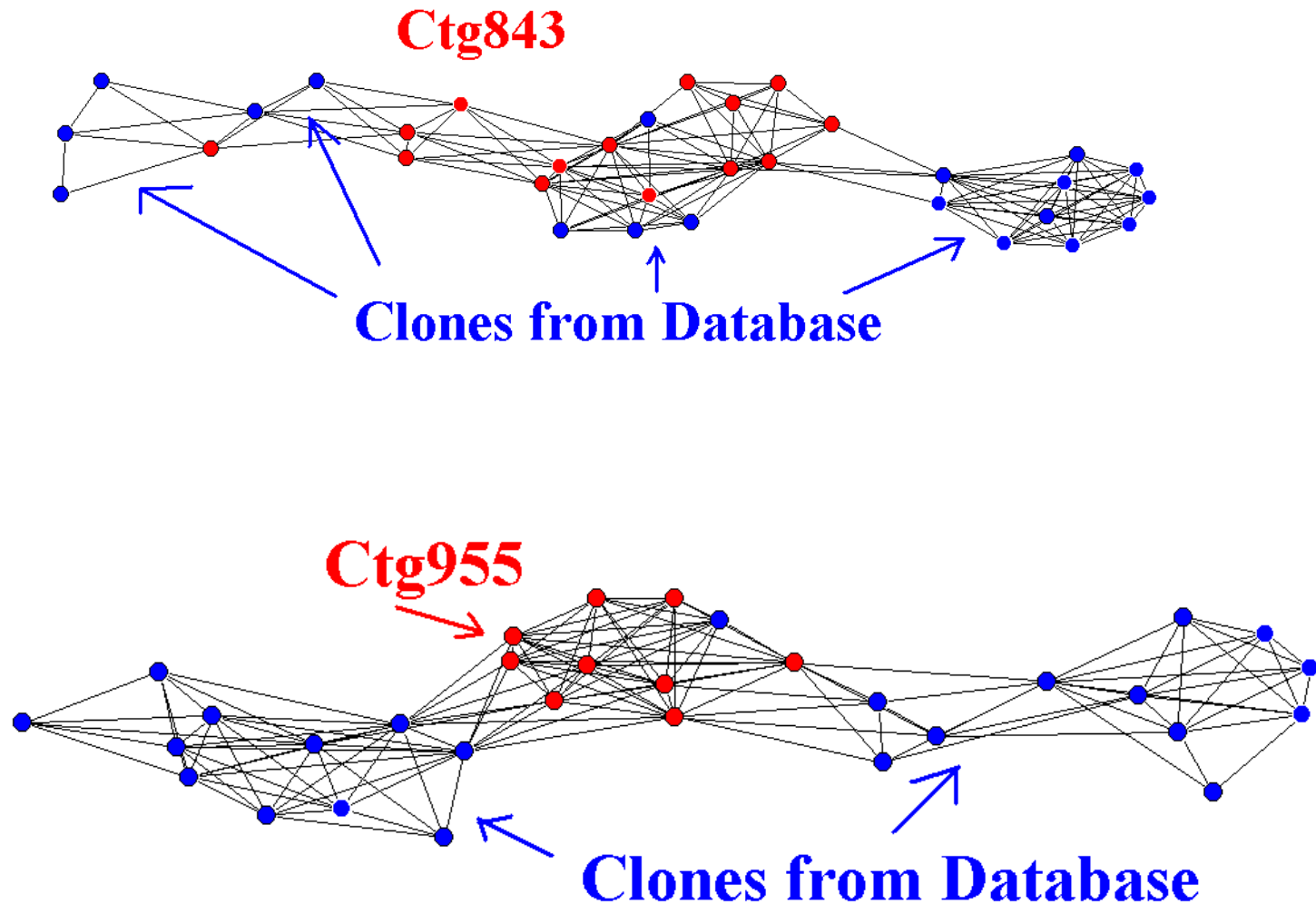
Testing FPC contigs by using LTC



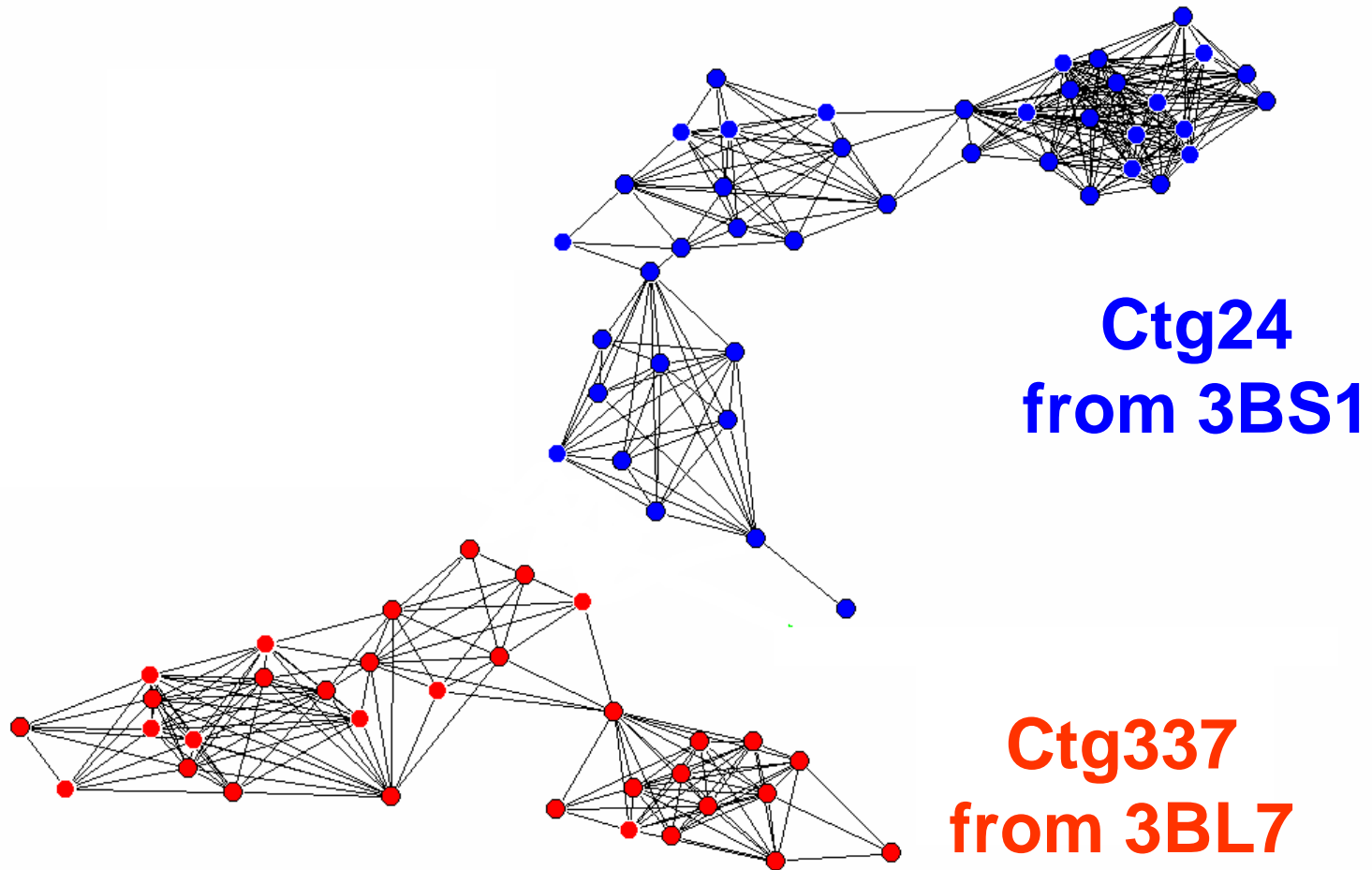
Testing FPC contigs by using LTC



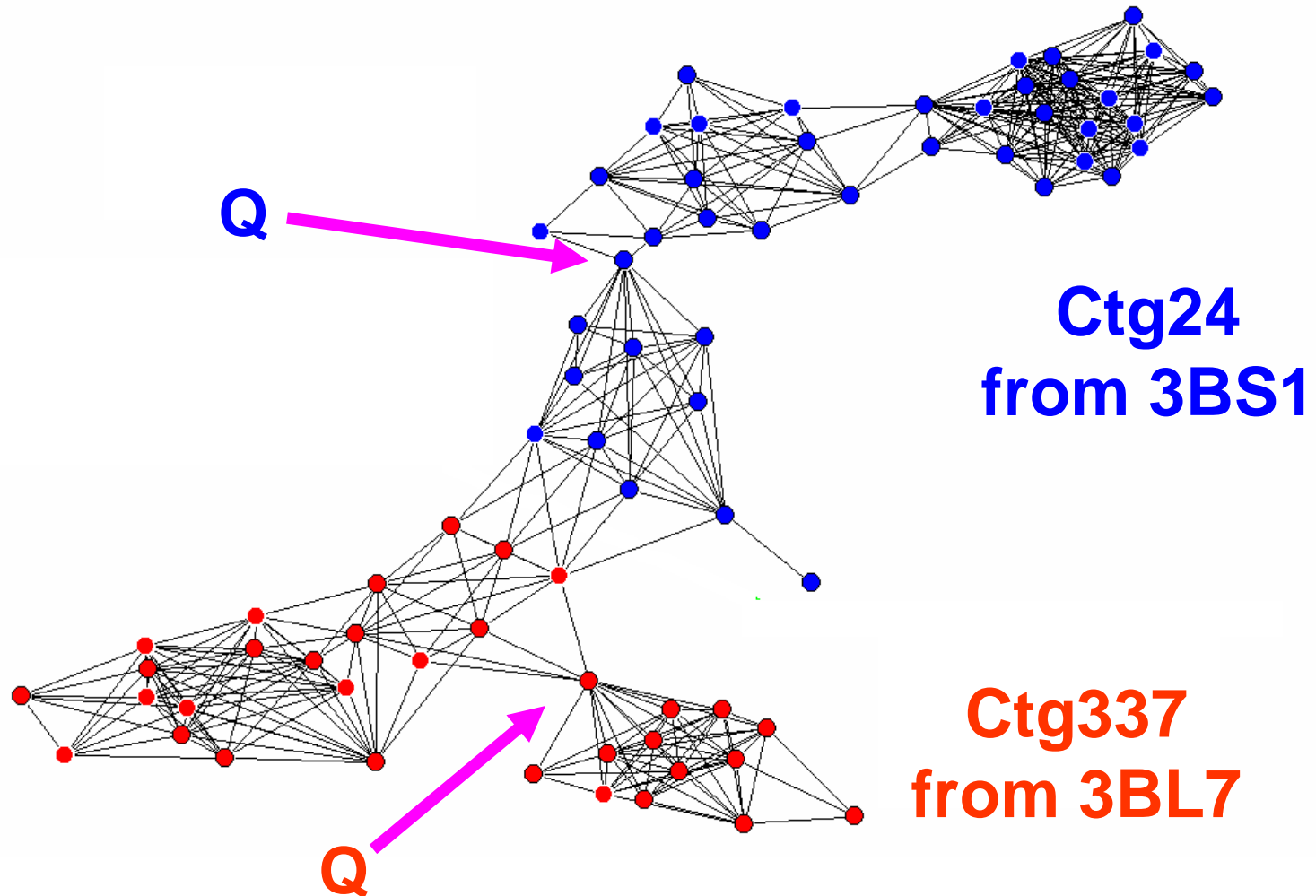
Elongation of FPC contigs



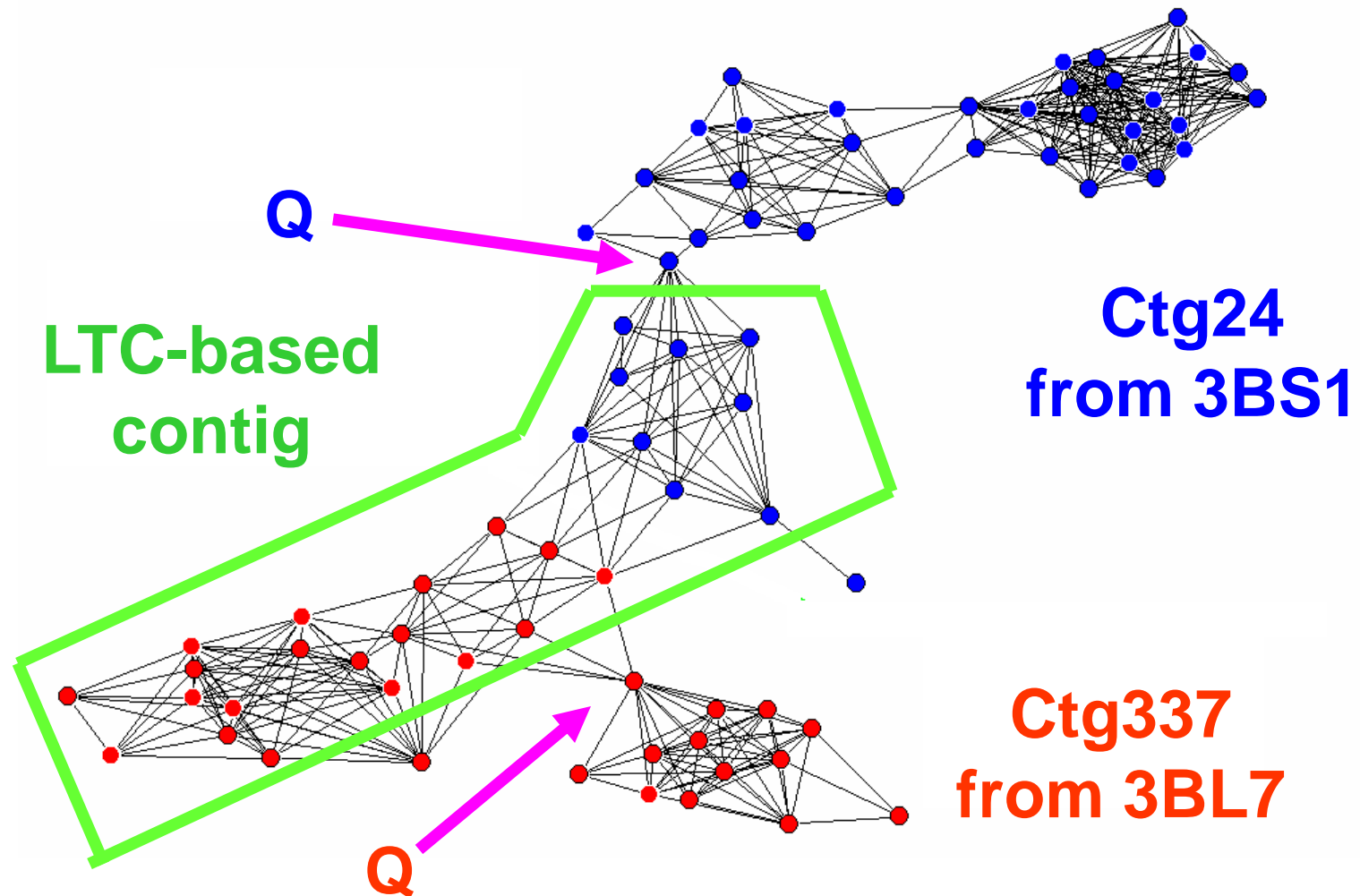
Reassembly of problematic FPC contigs



Reassembly of problematic FPC contigs



Reassembly of problematic FPC contigs



Some additional LTC tools

- Ordering of clones based on TSP
- Verification of the order by re-sampling
- Contig elongation and merging
- Identification of positive clones from a set of positive pools with errors
- Simulations based on sequenced genomes

Genome mapping as a Traveler Salesperson Problem (TSP)

<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>E</u>	<u>F</u>	<u>G</u>	<u>H</u>
a	b	c	d	e	f	g	h

How to choose the best (true) order, the one that gives the map of **minimal length**?

Order 1: a b c d e f g h k l m n l_1
Order 2: b a c d e f g h k l m n l_2

Order N: f c m h e a g n k l b d l_N



$n=60$ $N = 60!/2 \sim 3 \cdot 10^{56}$ orders

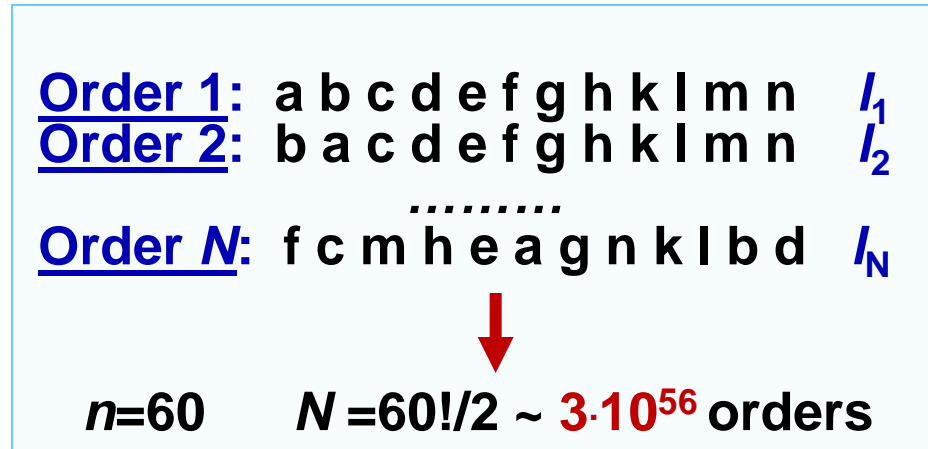
If n is small we can check all orders. If the data are exact we can choose the closest clones, then add the next closest, etc.
But the data are noisy. Need to check “all” orders !

Reduction to TSP

No exact solution exists to TSP. For practical situation various heuristic optimization methods were proposed, e.g.,
Evolutionary Strategy optimization

Mester et al. Genetics, 2003

ES algorithm for TSP based genome mapping



Consider order O_i as a ‘*genotype*’, and its ‘*fitness*’ as

$$w_i = l(O_i) = 1:l_i \quad (\text{or } -l_i)$$

“Progeny” is produced via *mutations* (changed orders).
A “child” replaces its parent if its fitness is higher. To order the contig we need only the p -values of pair-wise clone overlap for all pairs of clones of the contig.

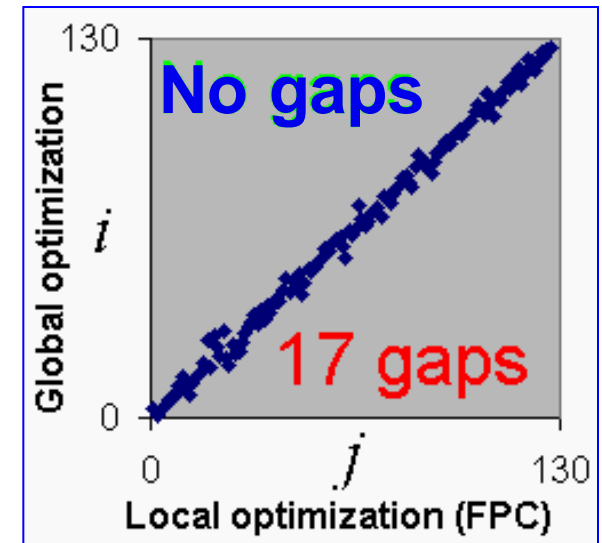
Contig ordering based on global optimization

(reducing clone ordering to TSP)

$$\text{TSP: } \sum_{j=1}^{n-1} d_{i(j), i(j+1)} \xrightarrow{F: j \leftrightarrow i} \min$$

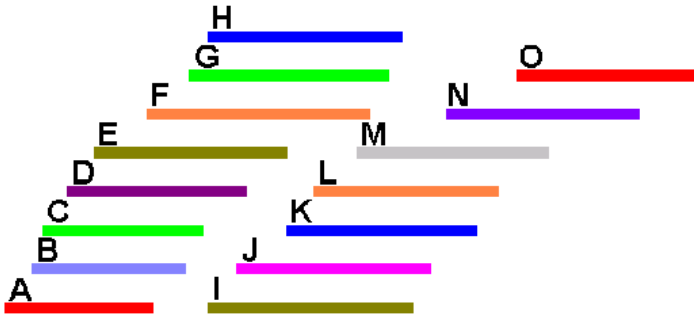
$$d_{q1, q2} = 1000 - (-\ln P_{q1, q2}) \cdot \mathbf{1}\{P_{q1, q2} < P_0\}$$

Less gaps: each pair of adjacent clones in the contig is significantly overlapped



Jackknife re-sampling for order verification

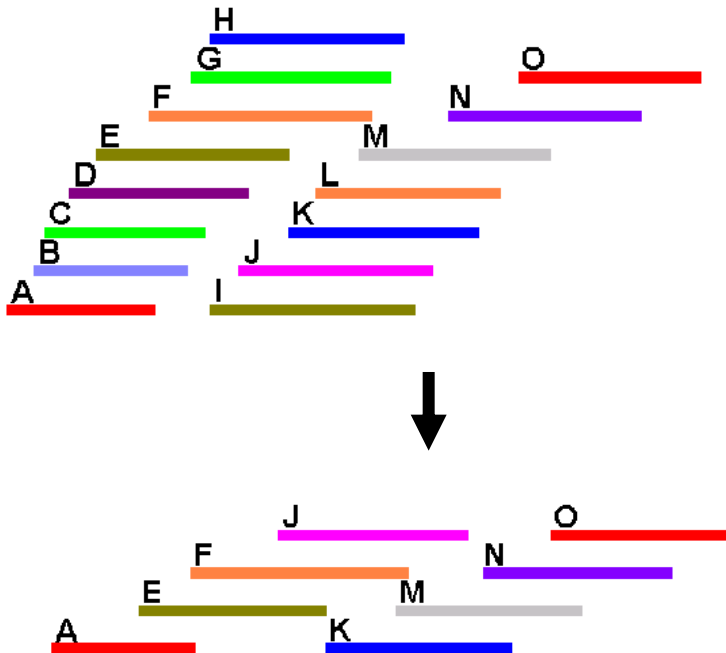
Excluding parallel clones allows constructing a stable "skeleton" map and specifying coordinates of all clones relative to this map.



			1	0.03															
		1		0.97															
	0.03	0.97			1														
				1		1													
					1		0.99	0.01											
						0.99		0.99	0.01	0.01									
						0.01	0.99		1										
							0.01	1		0.99									
								0.01	0.99		1								
										1		1							
											1		1						
												1		0.72	0.28				
													0.72		1	0.28			
														0.28	1		0.72		
															0.28	0.72			1
																		1	

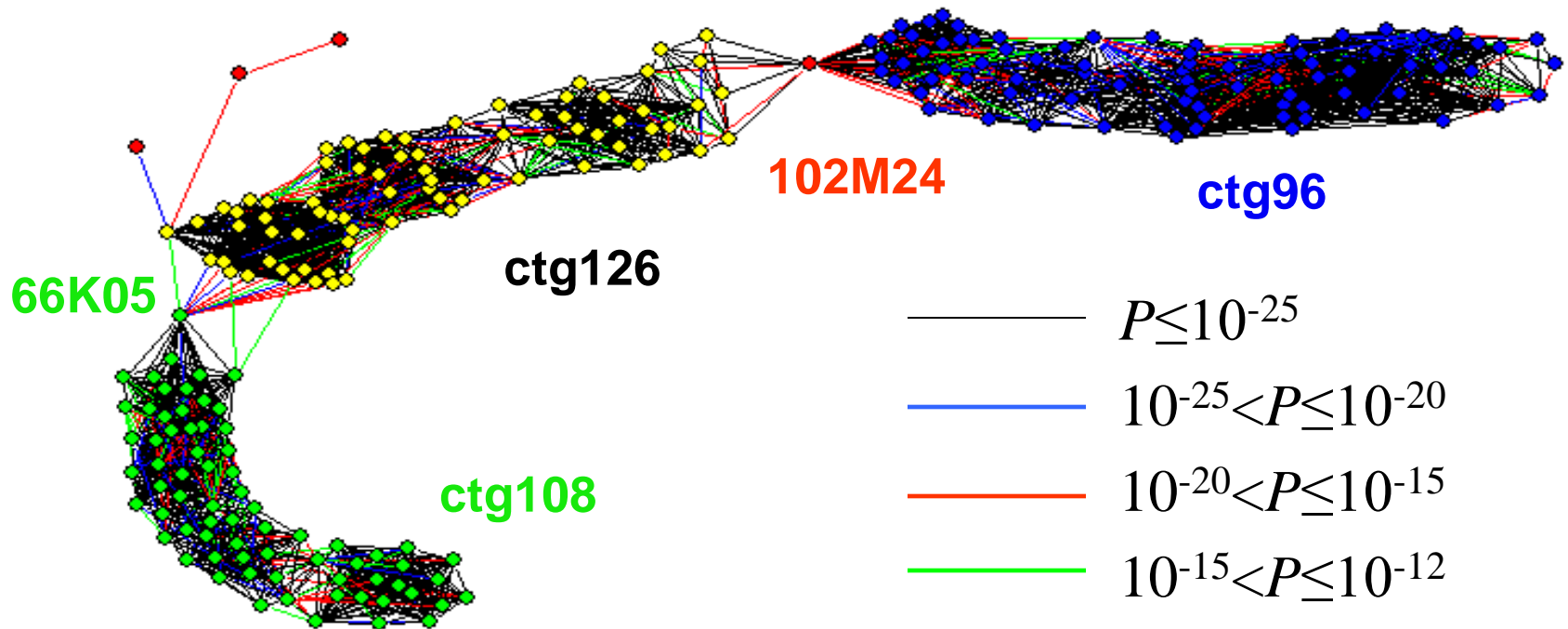
Jackknife re-sampling for order verification

Excluding parallel clones allows constructing a stable "skeleton" map and specifying coordinates of all clones relative to this map.



The diagram illustrates the construction of a 10x10 matrix A from a 5x5 matrix B . Matrix B is a lower triangular matrix with 1s on the diagonal and 0.03 in the top-left corner. Matrix A is a 10x10 matrix with 1s on the diagonal and 0.03 in the top-left corner. An arrow points from matrix B to matrix A , indicating that A is constructed from B .

End-to-end contig merging



Some results

LTC was used for

- Contig assembly and MTP selection for wheat 1BS, 1BL, 1AL, 5AS, 7BL, 7BS, 7AL, 7AS (with some assistance of HU group)
- Contig analysis, MTP selection and re-selection (with maximal using of already sequenced clones) in barley
- Alternative contig assembly and MTP selection for 1AS and 3B

Some results

1BS assembly: **FPC vs. LTC**

In total 49,412 clones	FPC	LTC
Contigs with ≥ 6 clones	517	385
Clones in contigs	33,262	33,912
Mean clones/contig	64.3	88.1
Clones in MTPs	3,647	3,827
Coverage by MTP	270 Mb (86%)	283 Mb (90%)

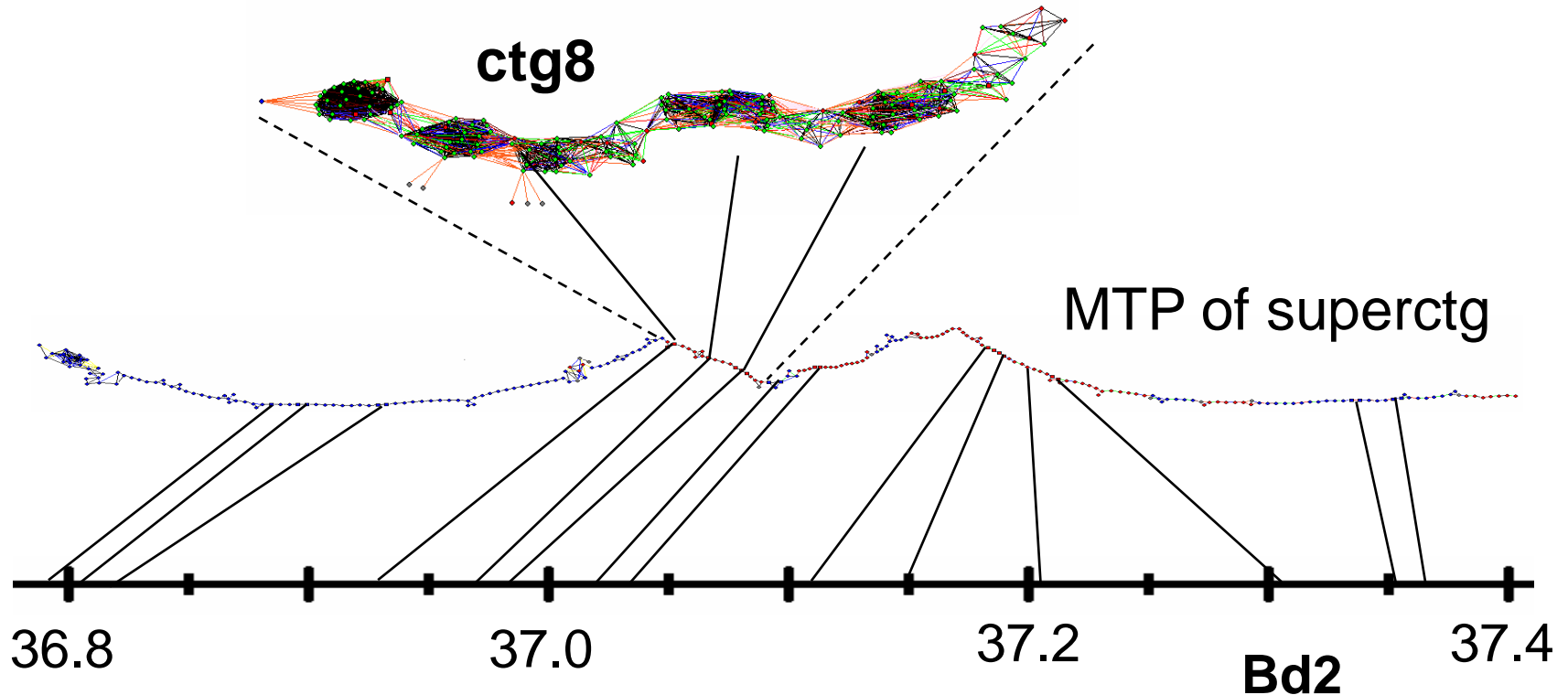
Average clone size 113 kb, coverage ~15

MTPs were constructed by LTC using 10^{-25} cutoff

Some results

- Construction of long (up to 20 Mbp) supercontigs for 1BS
- Identification of positive clones from transcriptomic and PCR experiments with 3D pools for 1BS
- Anchoring of supercontigs to maps of relative genomes (1BS to wheat genetic map and bin map, 1H to barley genetic map, and sequenced genomes of *Brachy* rice and sorghum)

A well anchored long 1BS supercontig



A supercontig with 2,061 clones covering ~17.8 Mbp of wheat 1BS anchored to 0.6 Mb of Bd2 by 14 markers.

Implementation

Assembly BAC contigs with LTC program

Input data

- a) HICF fingerprinting: lists of bands for each clone
- b) STS markers: lists of BACs with positive reactions
- c) WGP DNA-tag sequences: lists of BACs where exact tag was detected

Data preparation: BAC library

BAC length

- Short → Assembly more complicated
- Long → Identifying errors is difficult
- IWGSC: average ~100 to 200 kbp

Coverage:

- Low → gaps, less clones proven by parallels
- High → expensive
- IWGSC: average ~ x10 – x20

Names of clones:

clone_name = library_name + plate + well

Data preparation: BAC library

TaaCsp3DLhA_0023F15

- *Triticum aestivum* subspecies *aestivum*
- Chinese Spring
- the chromosome 3D, arm L
- h → HindIII enzyme used for the library construction
- A is the library code (first library)
- Plate 23
- Well in row F and column 15

Data preparation: BAC fingerprinting

IWGSC: HICF fingerprinting for clones

- Four enzymes
- ~1800 distinguishable bands of 50-500 bp (tolerance =0.4)
- Resolution: one band per ~1.1 kbp
 - ~100 bands per clone
 - some bands are missed

AB3730 sequencer → *.fsa file for each clone

Data preparation: input files

GeneMapper

*.fsa files → text table data

FPB

- Cleaning from background
- Removing from putative contaminations
- Renaming of clones to fit the FPC limitations
- Converting data into FPC format (*.sizes)

→ Folder of *.sizes files

Data preparation: main parameters

By FPB (to fit the FPC limitations):

- band size \rightarrow integer
- several enzymes \rightarrow one-dimensional array

IWGSC: $k=4$ enzymes, $b_{\min}=50$ bp, $b_{\max}=500$ bp,
 $s=30$

Total gel length $L=k \times b_{\max} \times s=60,000$

Tolerance $t=0.4 \times s=12$

$N_{\text{bands_Sulston}}=k \times (b_{\max}-b_{\min}) \times s / 2t=2,250$ bands

Data preparation: *.sizes files

CloneExample_014A15 **3** Gel

70

120

300

-1

CloneExample_018B21 **4** Gel

75

200

250

275

-1

LTC running: starting

- Run LTC_beta.exe
- Set main parameters
- Import HICF data from *.sizes files
 - Print report about band distribution
 - Print report about number of bands in clones
 - Save data in LTC-specific format
- Set the most liberal cutoff stringency
- Create the net of significant clone overlaps
 - Save the net in LTC-specific format

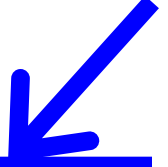
LTC running: starting

• Run LTC between

The stringency of cutoff should depend on data quality and size:

e.g., 10^{-12} for wheat chromosomes
(IWGSC, $\sim 10^5$ clones),
and 10^{-8} for STS or WGP tag data
($\sim 10^4$ clones).

- Save data in LTC-specific format

- Set the most liberal cutoff stringency 
- Create the net of significant clone overlaps
 - Save the net in LTC-specific format

Data example

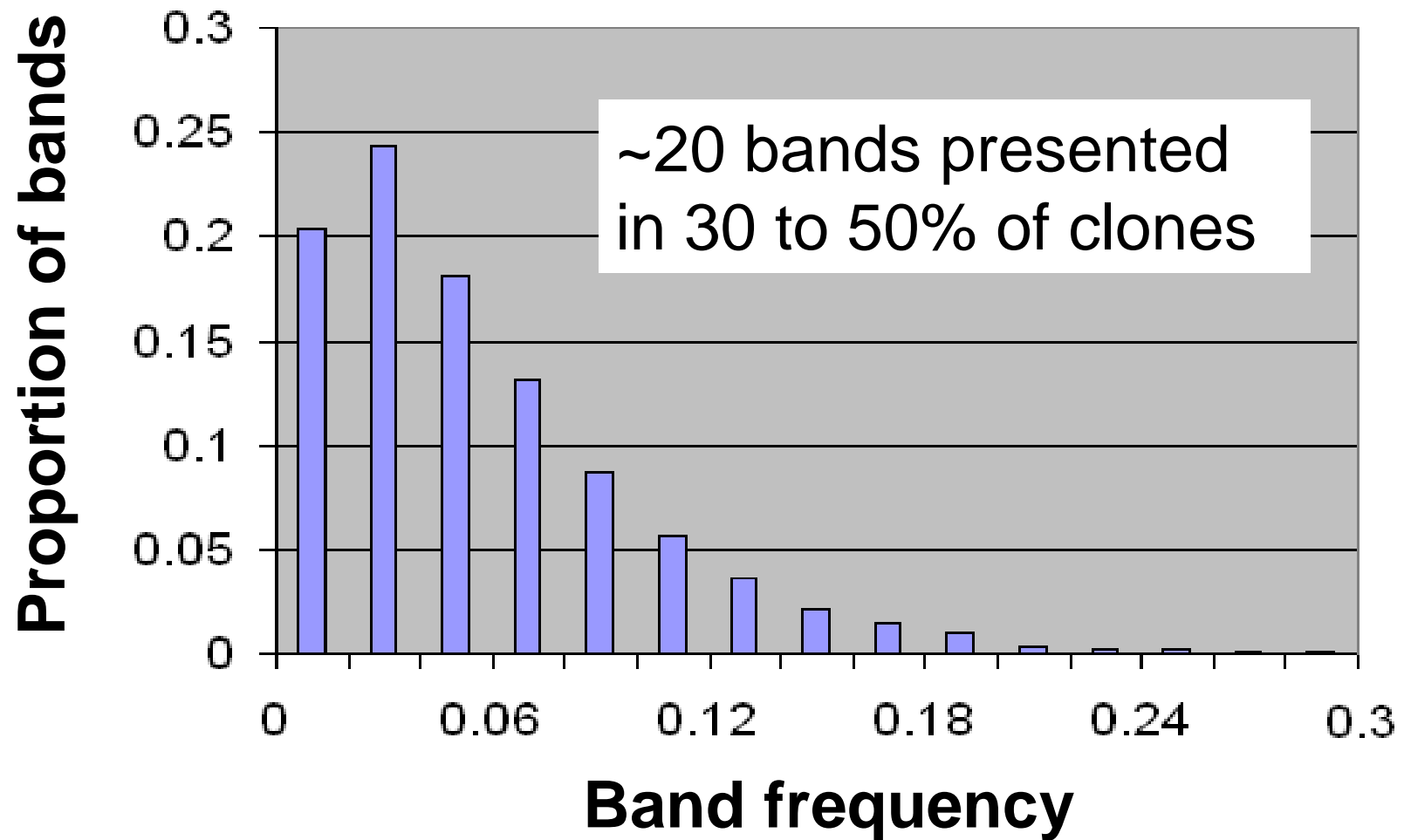
Simulated BAC-library based on published sequence of **maize** chromosome 1:

- 915 HICF-fingerprinted clones
- Coverage is about 12.0
- Simulated errors: 5% chimerical clones, 5% missed bands, 5% bands scored with errors higher than double tolerance

(Importance: The answer is known!)

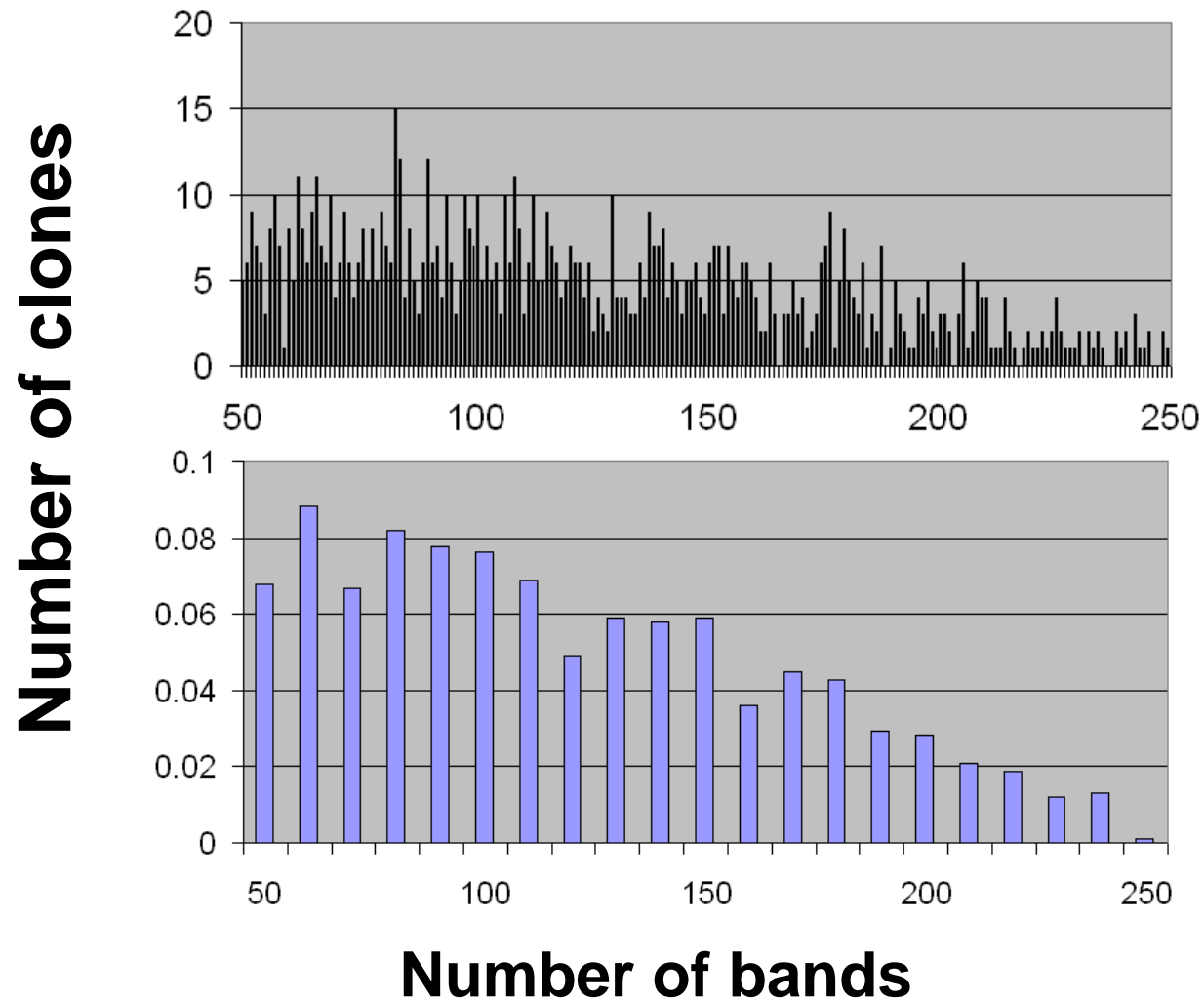
Let us start →

Basic statistics: Band distribution



(proportion of clones containing band)

Basic statistics: number of bands in clones

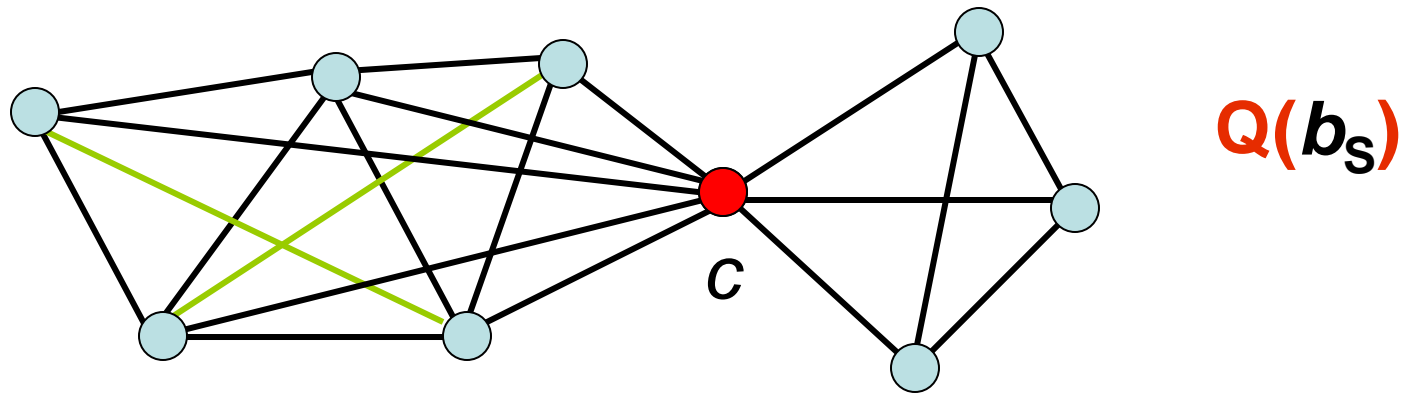


Detecting Q-clones by using two cutoffs

Stringent cutoff $\mathbf{b_s}$ (say, 10^{-25})

A more liberal cutoff $\mathbf{b_L}$ (say, 10^{-15})

For each clone c :



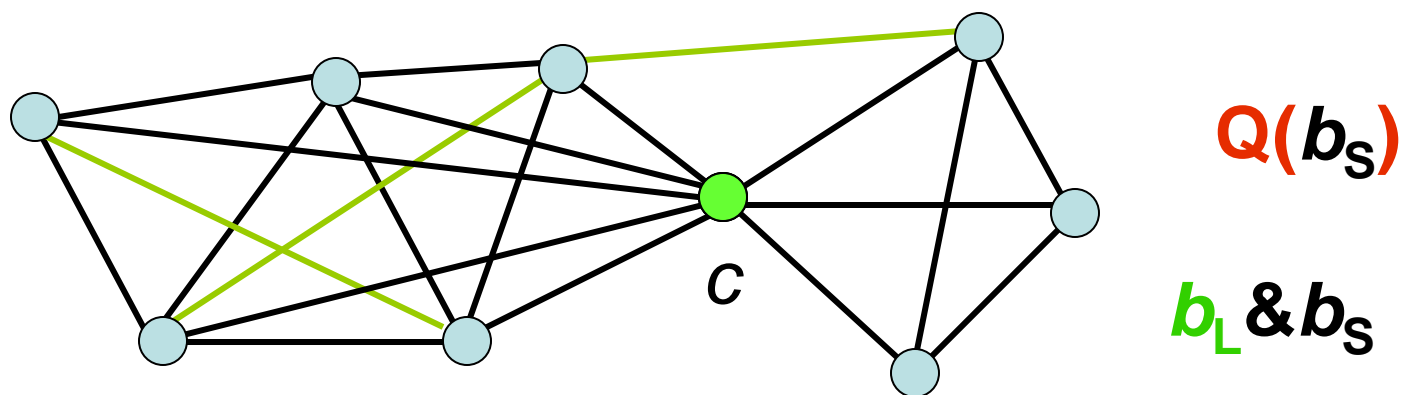
Q-clone \equiv $Q(\mathbf{b_L} \& \mathbf{b_s})$

Detecting Q-clones by using two cutoffs

Stringent cutoff $\mathbf{b_s}$ (say, 10^{-25})

A more liberal cutoff $\mathbf{b_L}$ (say, 10^{-15})

For each clone c :

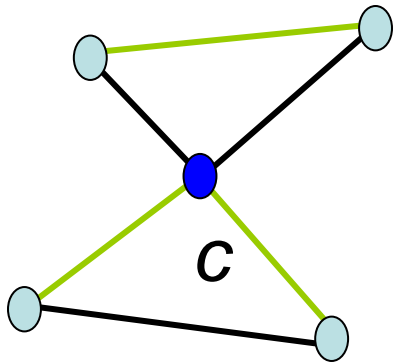


Comparison of Q-clones

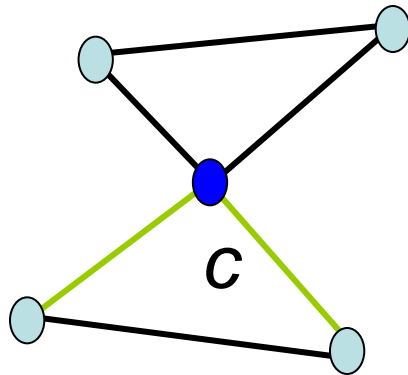
Liberal cutoff b_L (say 10^{-15})

More stringent cutoff b_S (say 10^{-25})

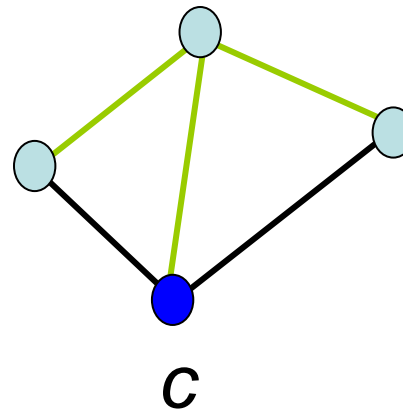
$b_S \rightarrow b_S \text{ \& } b_L$



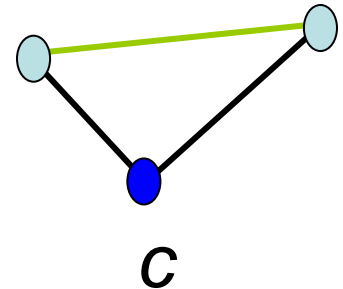
$Q(b_L),$
 $b_S \text{ \& } b_L,$
 $Q(b_S)$



$Q(b_L),$
 $b_S \text{ \& } b_L,$
 b_S



$b_L,$
 $Q(b_S \text{ \& } b_L),$
 $Q(b_S)$



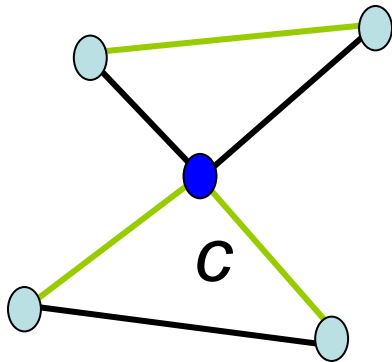
$b_L,$
 $b_S \text{ \& } b_L,$
 $Q(b_S)$

Comparison of Q-clones

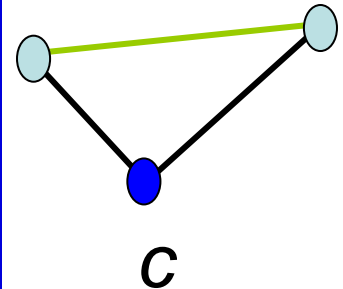
Liberal cutoff b_L (say 10^{-15})

More stringent cutoff b_S (say 10^{-25})

$b_S \rightarrow b_S \ \& \ b_L$



Conclusion: to be qualified as Q-clone depends on cutoff selection and on using additional cutoffs



$Q(b_L),$
 $b_S \ \& \ b_L,$
 $Q(b_S)$

$Q(b_L),$
 $b_S \ \& \ b_L,$
 b_S

$b_L,$
 $Q(b_S \ \& \ b_L),$
 $Q(b_S)$

$b_L,$
 $b_S \ \& \ b_L,$
 $Q(b_S)$

LTC running: simplified semi-automated contig assembly

- (A) Temporal exclusion of Q-clones and Q-overlaps using a liberal and a more stringent cutoffs
- (B) Adaptive clustering based on increasing cutoff stringency
- (C) If some of resulted clusters are **non-linear** or **too large**:
 - Temporal excluding of clones causing branching (for reasonable size clusters)
 - Temporal exclusion of Q-clones using more stringent cutoff (for too large clusters)
 - Repeat adaptive clustering

Example: a large cluster with highly overlapped clones

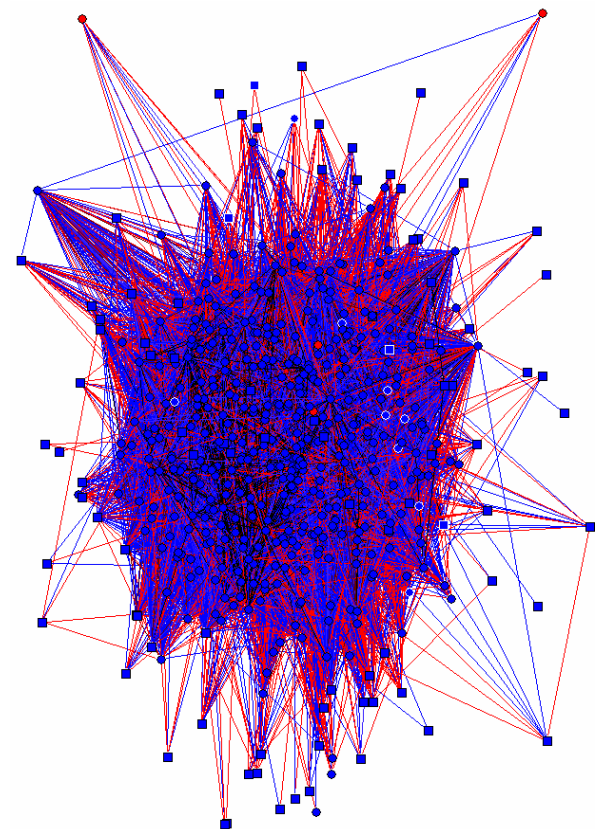
1BS: after exclusion of Q-clones with cutoff 10^{-30} we obtained cluster of 2110 clones.



Manual separation of several linear parts



Cluster with 1218 clones with average ~ 870 highly significant overlaps (cutoff 10^{-50}) per clone



Q-clones and Q-overlaps: statistics

Q-overlaps (cutoff 10^{-16}):

Six Q-overlaps found, $\min(p\text{-value})=10^{-20}$

Q-clones (cutoffs 10^{-16} and 10^{-25}):

17 Q-clones found:

one Q-clone is not chimerical

over 16 are chimerical

27 of simulated chimerical still not found

Not-detected chimerical clones

27 of simulated chimerical clones not found:

Adaptive clustering:

- Only three chimerical contigs

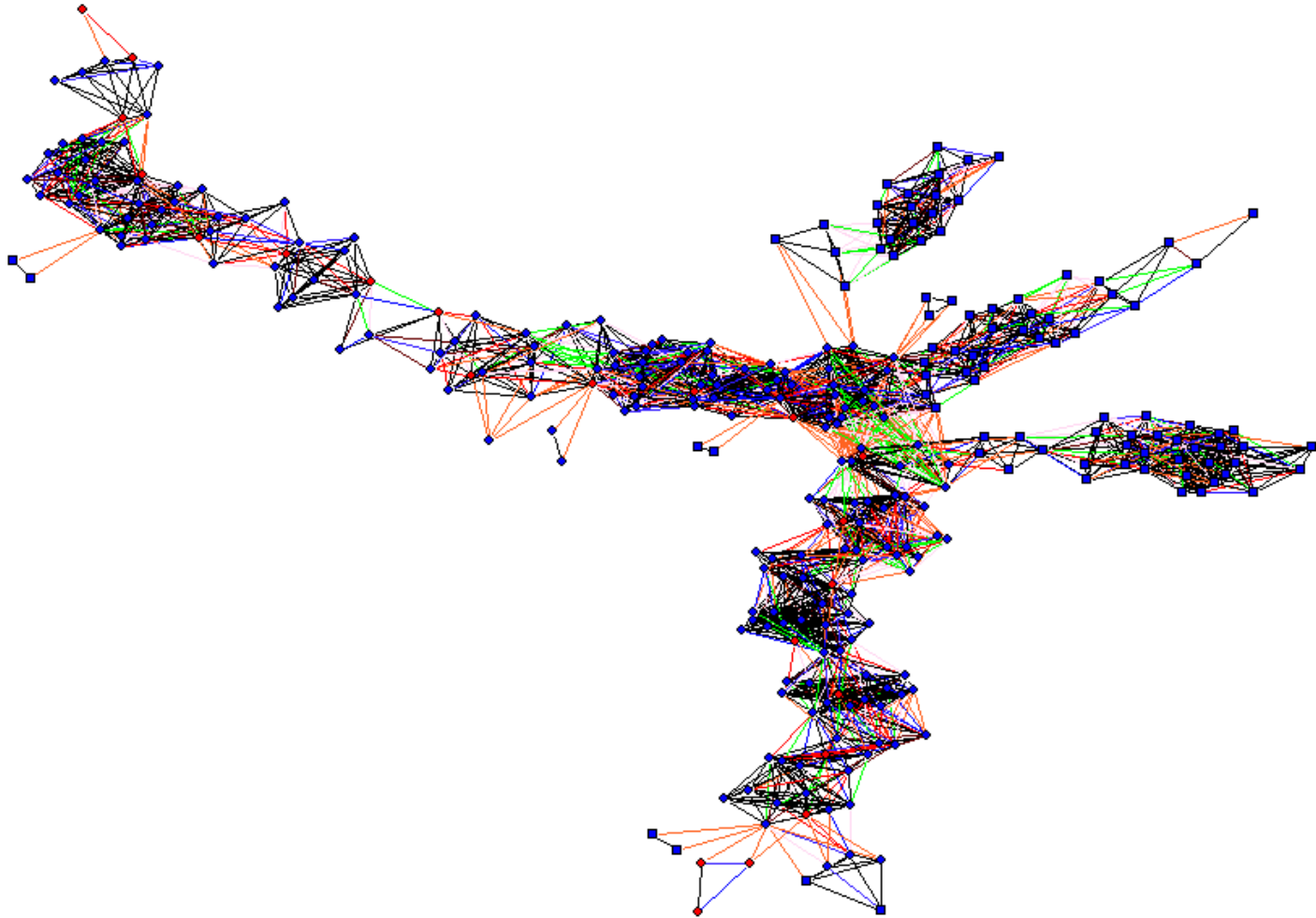
- Only one chimerical clone in connection of chimerical contigs

- Other 26 chimerical clones do not connect contigs at the resulted cutoffs

Resulted contigs and clusters

cluster	n_clones	cutoff	diam	width	n_gaps	n_rank_M1
cl_2	27	16	4	1	0	0
cl_3	181	16	17	1	1	0
cl_4	49	16	7	1	0	0
cl_6	16	16	4	1	0	0
cl_8	28	16	3	1	0	0
cl_9	123	16	15	1	0	0
cl_10	6	16	2	1	0	0
cl_11	29	16	4	1	0	0
cl_12	11	16	2	1	0	0
cl_13	32	16	4	1	0	0
cl_15	28	16	3	1	0	0
cl_17	16	16	3	1	0	0
cl_18	16	16	3	1	0	0
cl_20	19	16	3	1	0	0
cl_22	7	16	2	1	0	0
cl_41	22	19	5	1	0	0
cl_56	121	34	17	1	2	0
cl_57	140	34	22	1	0	0

Visualization of net of clone overlaps for a cluster (not contig!)



Additional clones (potential “bridges”)

- Q-clones
- Clones excluded from branching foci
- Clones with rank ≥ 2 relative to MTP
- Clones removed by clustering with more stringent cutoff

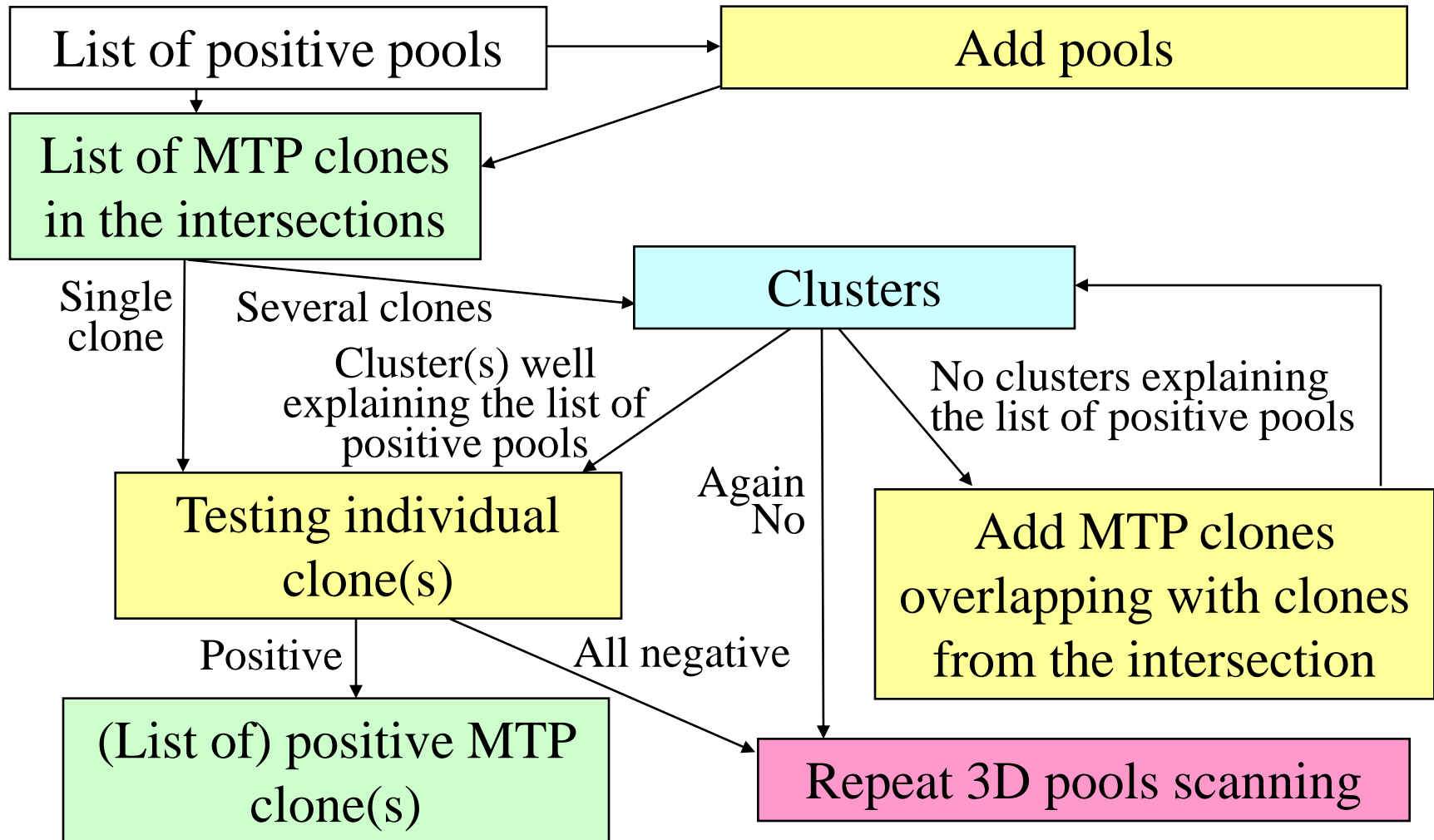
For these clones (together with MTP clones)

- Pooling to simplify scanning for marker presence
- BAC-end-sequencing [?]
- Sequencing [??]

At this stage contig assembly is finished

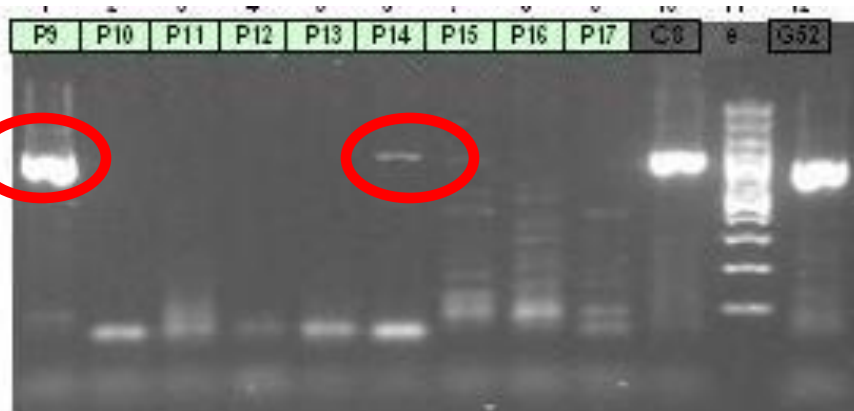
Additional (post-assembly) tools

Cost-effective search of positive MTP clones using 3D-pools with scoring errors

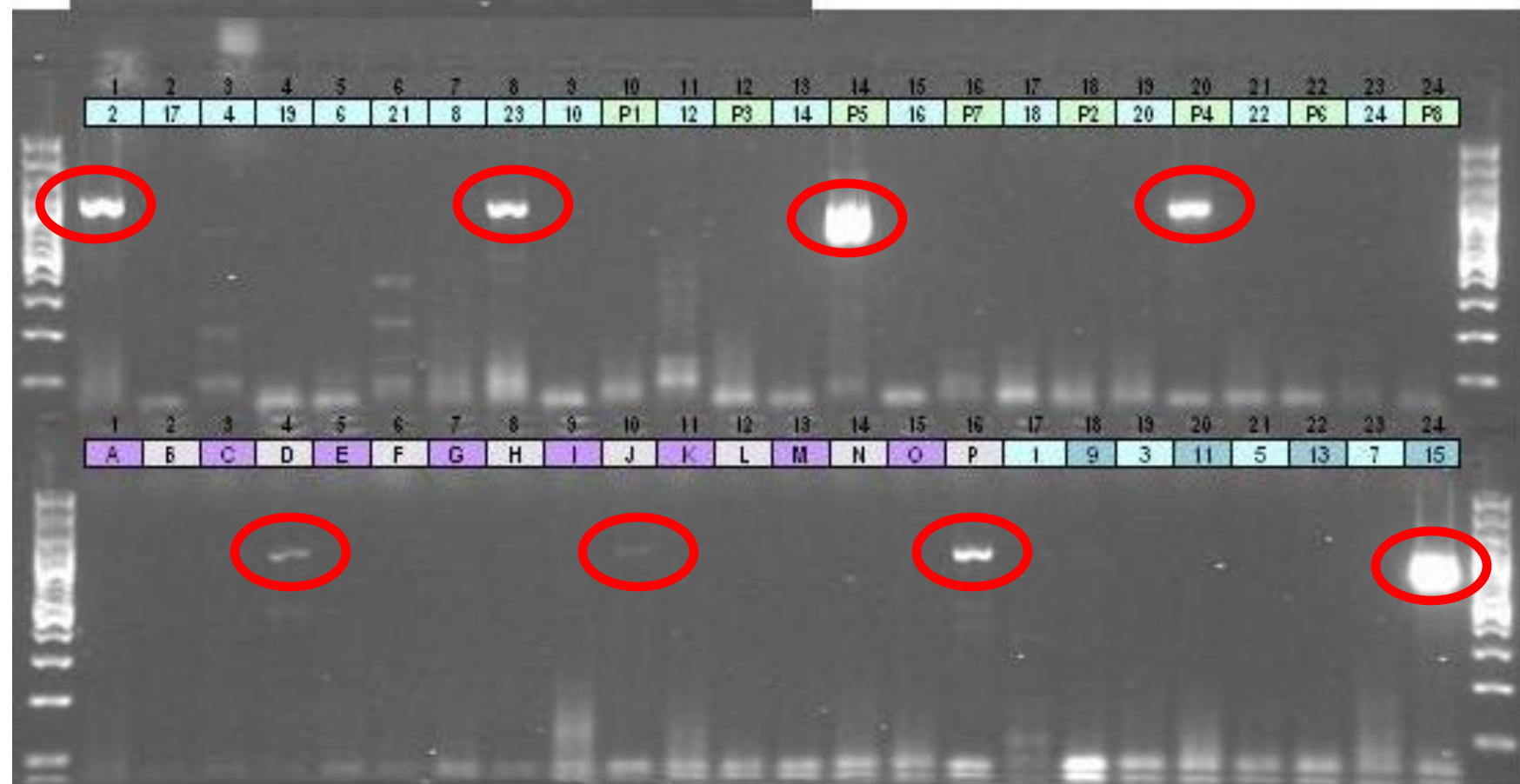


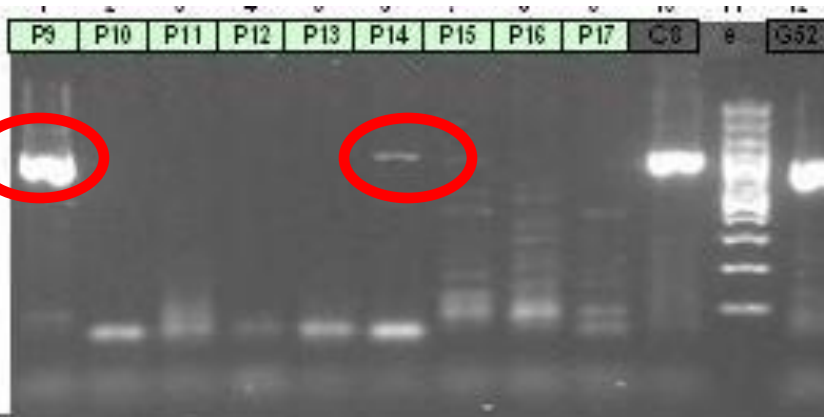
Identification of positive clones from the list of positive 3D-pools

- Set basic parameters
- Input HICF data
- Set cutoff
- Create/input the net of significant clone overlaps
- Input contig assembly from *.fpc file
- Input new (pool) names for MTP clones
- Input list of positive pools



Plates: P4, P5, P9, P14?
Columns: 2, 15, 23
Rows: D, J?, P





Plates: P4, P5, P9, P14?
Columns: 2, 15, 23
Rows: D, J?, P

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
2	17	4	19	6	21	8	23	10	P1	12	P3	14	P5	16	P7	18	P2	20	P4	22	P6	24	P8

4 plates*3 rows*3 columns=36 candidate clones

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	1	9	3	11	5	13	7	15

Identification of positive clones from list of positive 3D-pools

Clustering of the 36 candidate clones (cutoff 10^{-15}):

- 29 singletons (3 pools of 10 positive),
- 2 clusters with two clones (explain 4 and 5 positive pools out of 10 positive)
- 1 cluster with three clones (9 pools of 10 positive → one positive pool p9 remains unexplained)

Adding clones (cutoff 10^{-15} , rank=1) and clustering:

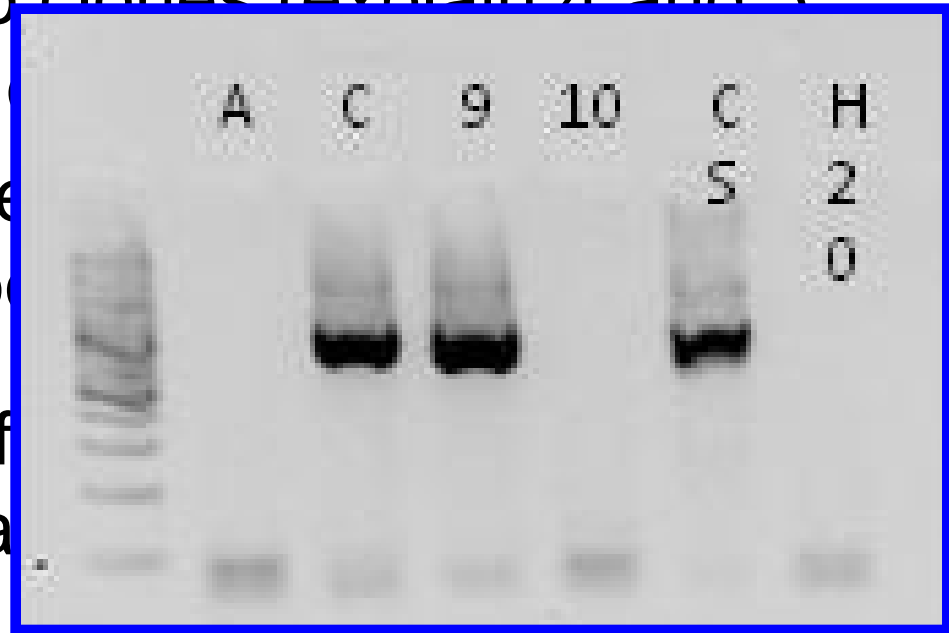
→ One cluster contains four clones explaining all of 10 positive pools, but two pools **C** and **9** are negative? → **need wet tests!**

Identification of positive clones from list of positive 3D-pools

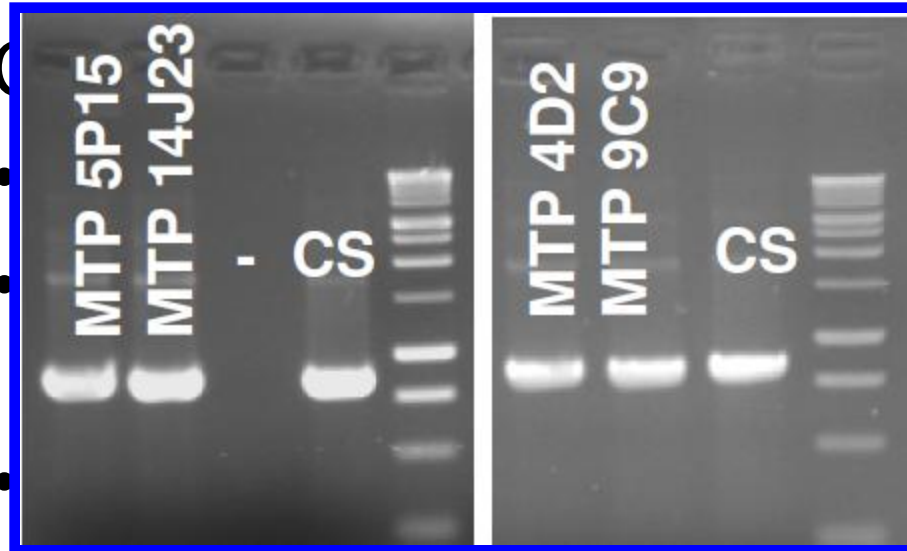
Clustering of the 36 candidate clones (cutoff 10^{-15}):

- 29 singletons (3 pools of 10 positive),
- 2 clusters with two clones (explain 4 and 5 positive pools out of 10)
- 1 cluster with three clones (explain 6 positive pools out of 10)
→ one positive pool

Adding clones (cutoff 10^{-15})
→ One cluster containing 3 clones (explain 10 positive pools, but two pools are negative? → **need wet tests!**)



Identification of positive clones from list of positive 3D-pools



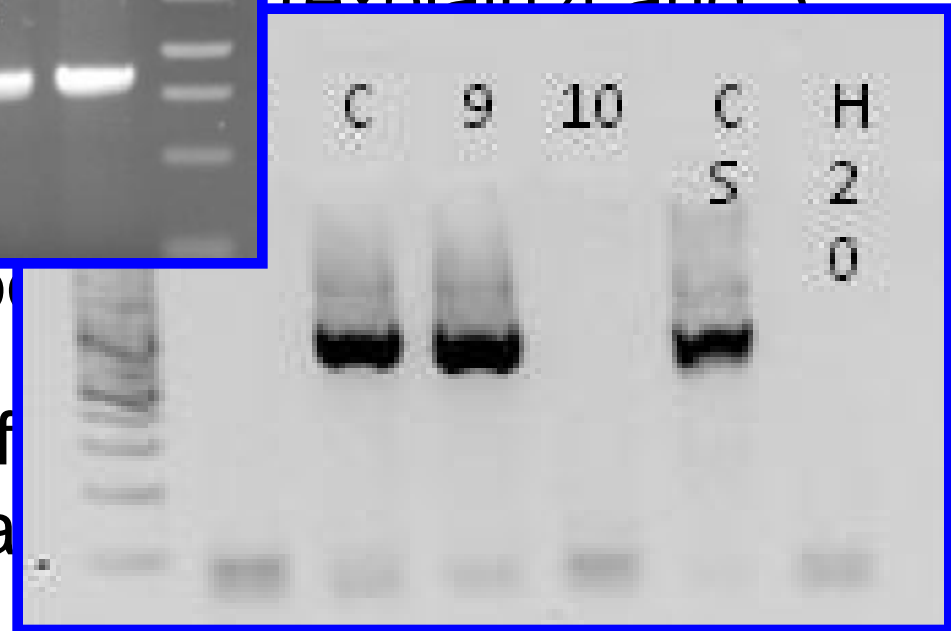
te clones (cutoff 10^{-15}):
 0 positive),
 (explain 4 and 5

→ one positive pool

Adding clones (cutoff

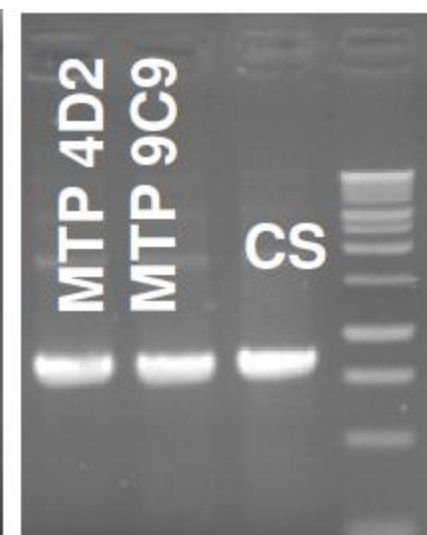
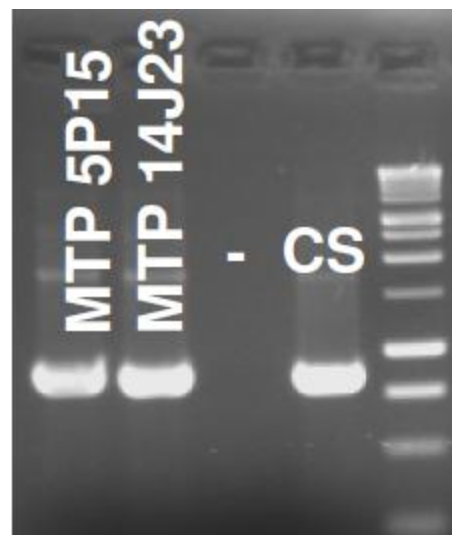
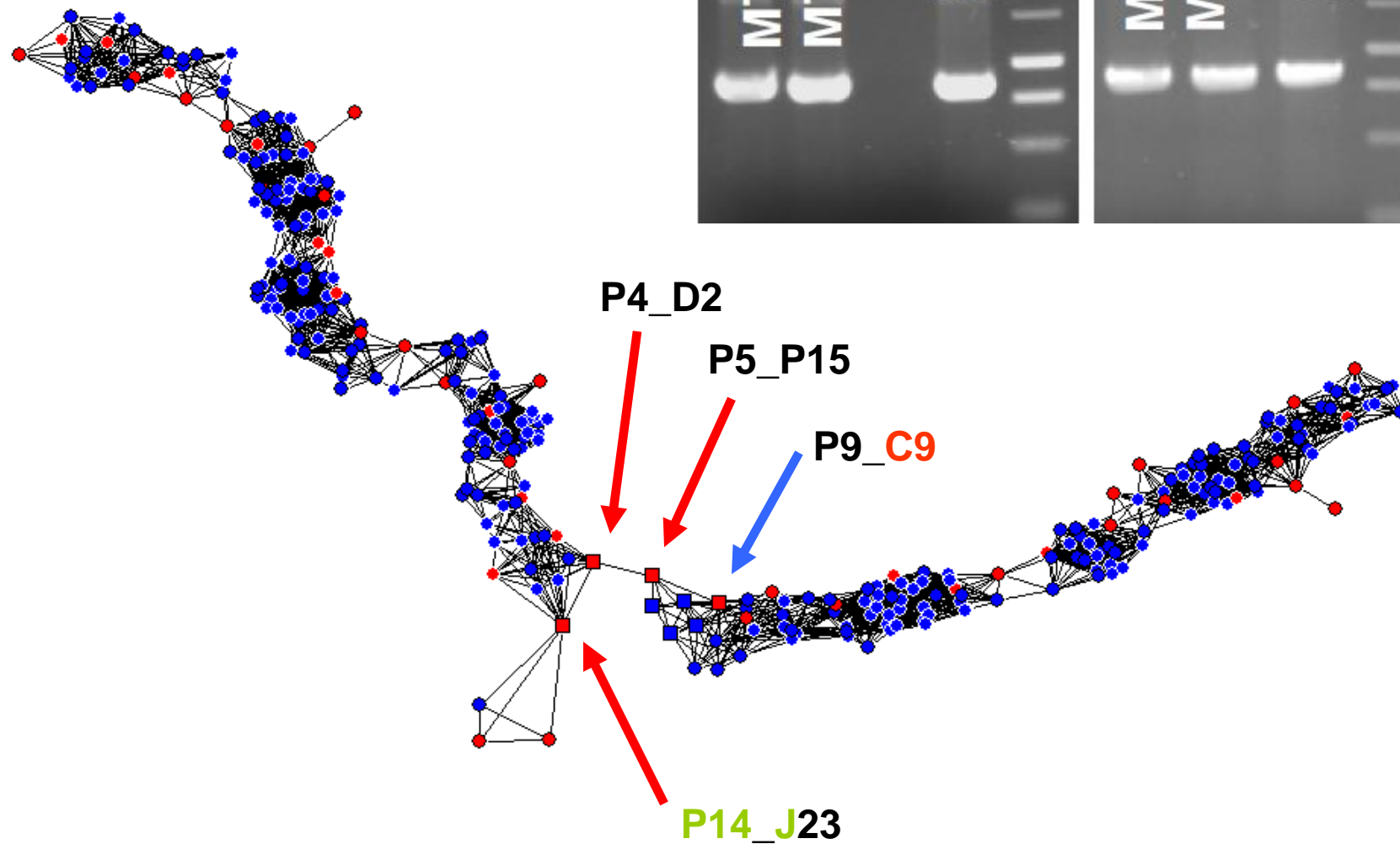
→ One cluster conta

10 positive pools, but two pools are negative? → **need wet tests!**



ve

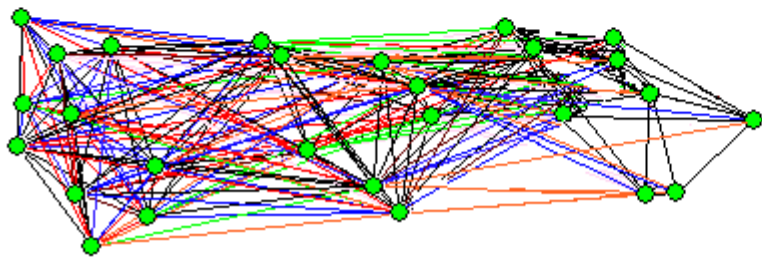
g:
 of



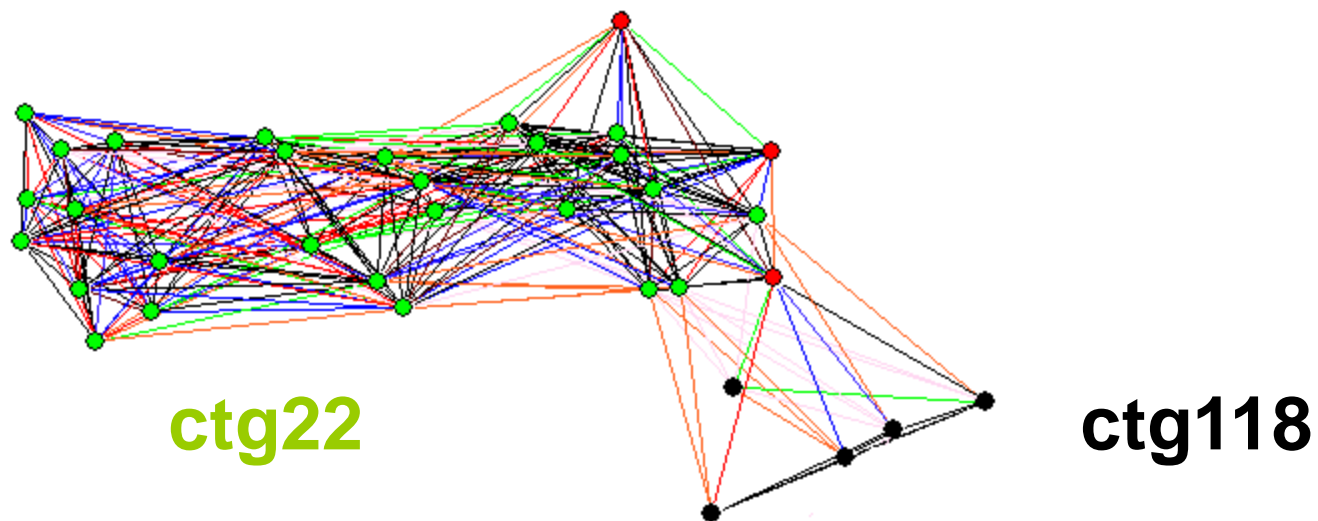
Contig elongation and merging

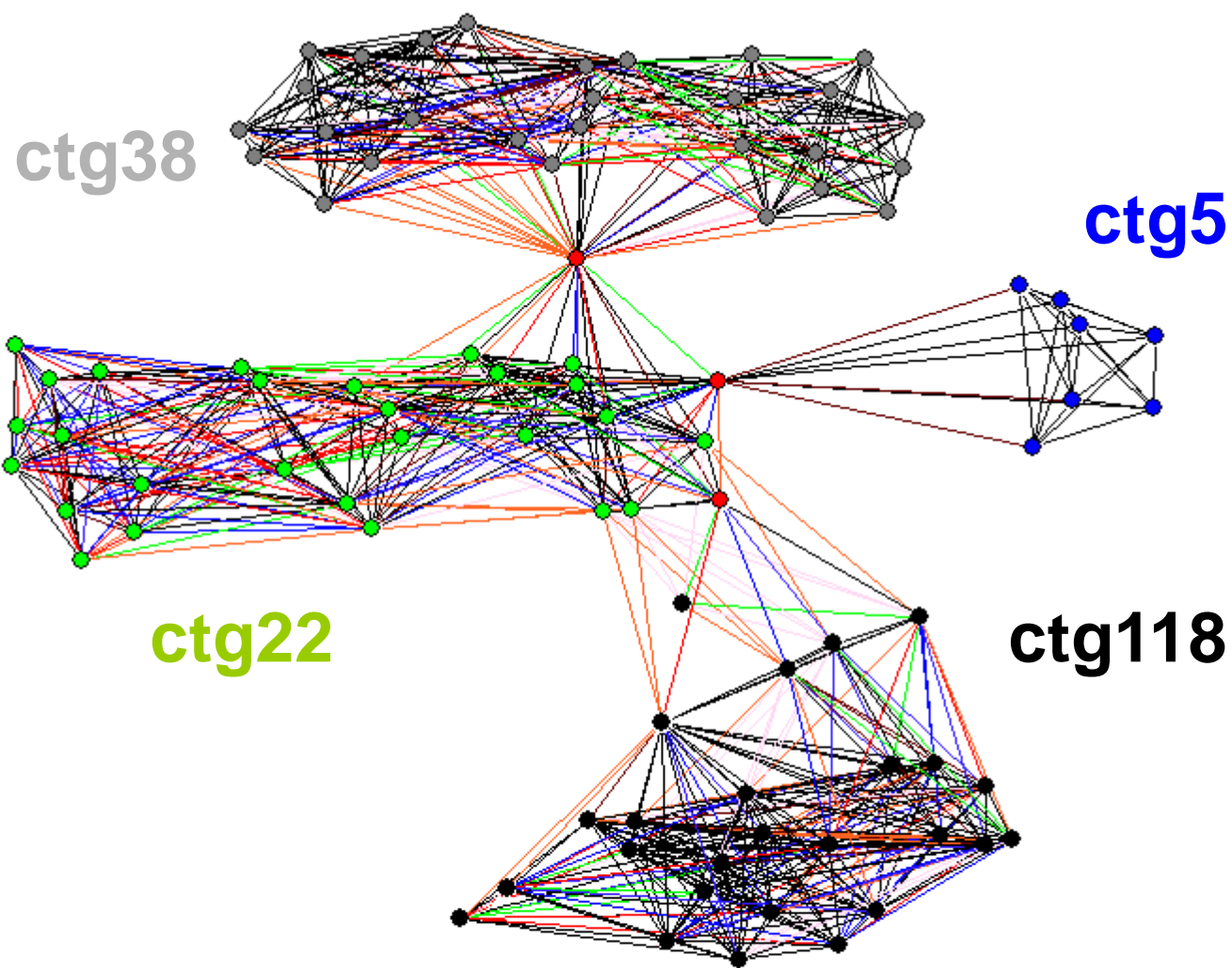
Contig elongation and merging

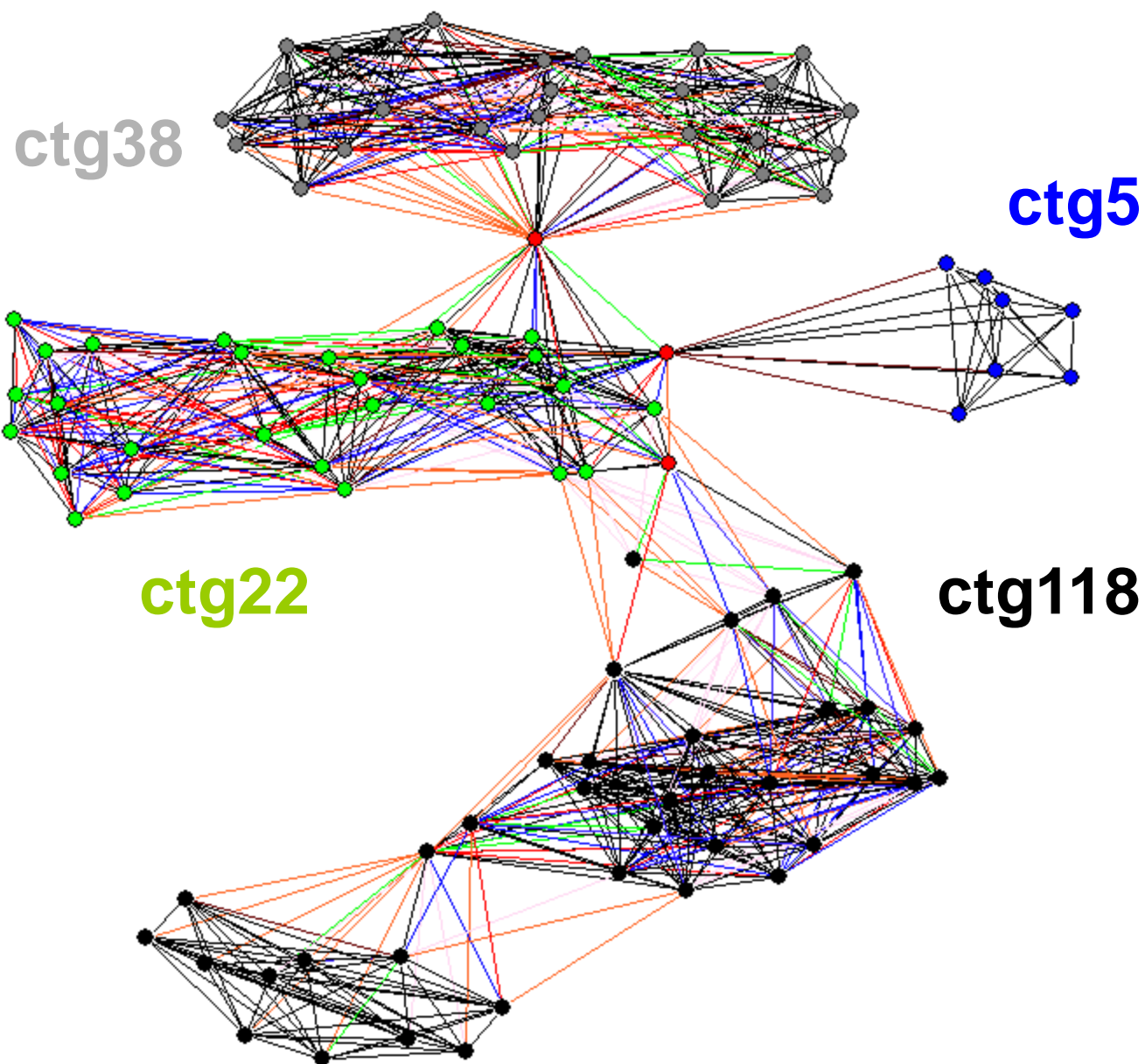
- Input HICF data
- Set cutoff
- Create/input the net of significant clone overlaps
- Input contig assembly from .fpc file
- Set a starting contig or supercontig

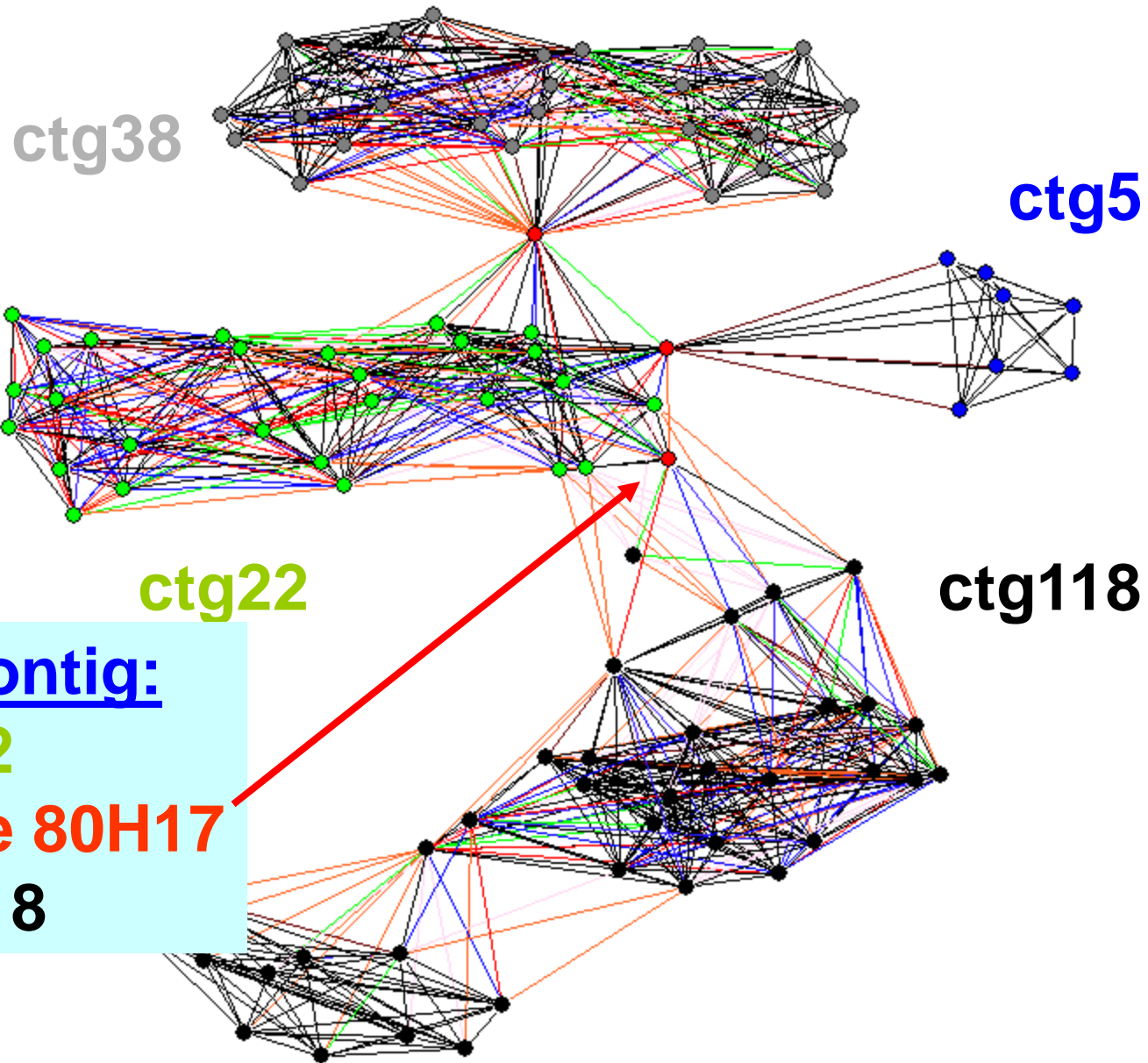


ctg22









Supercontig:

ctg22

clone 80H17

ctg118

Testing of FPC contig assembly

Testing of FPC based contig assembly

- Set basic parameters
- Input HICF data
- Set cutoff
- Create/input the net of significant clone overlaps
- Input contig assembly from .fpc file
- [optionally] Input MTP made by FPC

Detectable problems

- Non-connected contigs
- Contigs connected via Q-clones
- Non-linear contigs
- Non-significant overlap of adjacent MTP clones

Acknowledgements



- Zeev Frenkel
- Dina Raats
- Tamar Krugman
- Itay Dodek
- Elitsur Yaniv
- David Mester
- Hanan Sela
- Tzion Fahima
- Abraham Korol



- Vladimir Glickson



- Etienne Paux
- Romaine Phylippe
- Catherine Feuillet



UNIVERSITY OF ZURICH

- Thomas Wicker



- Alan Schulman

Thank you