### THE COMPLEXITY OF FINDING COMMON PARTITIONS OF GENOMES WITH PREDEFINED BLOCK SIZES

Manuel Lafond<sup>1</sup>, Adiesha Liyanage<sup>2</sup>, Binhai Zhu<sup>2</sup>, Peng Zou<sup>2</sup>

<sup>1</sup>*Université de Sherbrooke, Canada* <sup>2</sup>*Montana State University, USA* 

# **Synteny** : block of at least two genes that was conserved across multiple species.



Leptospira strains syntenies Image taken from Ramli et al.: <u>https://doi.org/10.3390/pathogens10091198</u>

# **Synteny** : block of at least two genes that was conserved across multiple species.



Human vs pig synteny map Image taken from Kim et al.: <u>https://doi.org/10.1186/1471-2164-13-711</u> **Synteny** : block of at least two genes that was conserved across multiple species.

To find syntenic blocks between two genomes *S* and *T*:

- Partition genes into homologous families
- Represent *S* and *T* as strings (each symbol = 1 gene family)
- Partition *S* and *T* into identical substrings (blocks)

### a a b a a b a a a b a b

### a b a b a a a b a a b a

# a a b a a b a <mark>a a b a b</mark>

# a b a b a a a b a a b a



Usual formulation: Minimum Common String Partition (MCSP)

Given strings *S*, *T*, split them into two identical (multi)sets of blocks, while minimizing the number of blocks.



Usual formulation: Minimum Common String Partition (MCSP)

Given strings *S*, *T*, split them into two identical (multi)sets of blocks, while minimizing the number of blocks.

### a a b a a b a a a b a b

# a b a b a a a b a a b a

Usual formulation: Minimum Common String Partition (MCSP)

Given strings *S*, *T*, split them into two identical (multi)sets of blocks, while minimizing the number of blocks.











Given two strings *S*, *T* and min block size *b*, **does there exist** a common partition of *S* and *T* in which the size of each block is at least *b*?

Given two strings *S*, *T* and min block size *b*, **does there exist** a common partition of *S* and *T* in which the size of each block is at least *b*?

#### The Min-Strip Recovery Problem

Given two strings *S*, *T* and min block size *b*, **delete a minimum number** of characters from the strings, so that they admit a common partition with blocks of size at least *b*.

#### The Min-Strip Recovery Problem

Given two strings *S*, *T* and min block size *b*, **delete a minimum number** of characters from the strings, so that they admit a common partition with blocks of size at least *b*.

b = 4

# a a b a a b a a a b a b

### a b a b b a a b a a b a

#### The Min-Strip Recovery Problem

Given two strings *S*, *T* and min block size *b*, **delete a minimum number** of characters from the strings, so that they admit a common partition with blocks of size at least *b*.



#### <u>Min Strip Recovery</u>

- Formulation in [Zheng, Zhu & Sankoff 2007]
- Heuristics based on Maximum Clique [Choi et al 2007]

											RICE										
1	+001	+003	+004	+006	+007	+008	+010	+013	+014	+016	+017	+019	+020	+021	+022	+023	+024	+028	+029	+032	+035
+037	+038	+039	+040	+041	+042	+043	+044	+049	+051	+052	+053	+054	+055	+056	+057	+058	+059	+060	+061	+062	+063
+064	+065	+067	+069	+070	+071	+072	+073	+076													
2	+080	+081	+083	+084	+085	+086	+087	+088	+089	+090	+092	+093	+094	+095	+096	+100	+102	+104	+105	+106	+109
+110	+111	+112	+114	+115	+117	+118	+119	+120	+121	+122	+123	+124	+125	+126	+127	+128	+129				
3	+131	+132	+133	+135	+136	+137	+138	+139	+140	+144	+145	+147	+148	+149	+150	+151	+152	+153	+154	+155	+156
+157	+158	+159	+160	+161	+162	+163	+164	+165	+169	+170	+172	+175	+177	+179	+181	+182	+183	+184	+185	+187	+188
+191	+192	+193	+194	+195	+198	+200	+201	+202	+203												
4	+206	+207	+210	+211	+212	+214	+217	+218	+220	+222	+223	+224	+226	+229	+230	+234	+235	+237	+238		
5	+239	+242	+243	+244	+245	+246	+249	+250	+251	+252	+260	+261	+262	+263	+265	+268	+269	+272	+273	+274	+276
+277	+279																				
6	+280	+281	+283	+284	+285	+286	+288	+289	+290	+291	+292	+293	+295	+299	+304	+305	+307	+308	+310	+311	+313
+314	+315	+317	+318	+320	+321	+322	+323	+324	+325	+326				_						_	
7	+327	+328	+329	+330	+332	+334	+336	+337	+338	+340	+342	+343	+344	+347	+350	+351	+352	+355	+356	+357	
8	+358	+362	+363	+365	+367	+371	+372	+373	+374	+377	+378	+379	+380	+381	+382	+383	+386	+387			
9	+388	+392	+393	+394	+395	+396	+397	+398	+399	+402	+403	+405	+409	+410	+413						
10	+416	+418	+420	+423	+425	+426	+427	+429	-												
11	+431	+432	+434	+435	+436	+438	+439	+440	+442	+443	+447	+448	+449	+450	+452	+454	+455	+456			
12	+460	+464	+466	+470	+474	+477															
										s	ORGHU	м									
										s	ORGHU	м									
A	-013	-010	-008	-007	-006	-004	-003	-001	+014	s +016	ORGHU	M +019	+020	+021	+022	+023	+024	+028	+029	+032	+035
<b>A</b> +037	-013 +038	-010 +039	-008 +040	-007 +041	-006 +042	-004 +043	-003 +044	-001 +049	+014 +051	+016 +054	ORGHU +017 +055	M +019 +056	+020 +057	+021 +058	+022 +059	+023	+024 +061	+028 +062	+029	+032	+035
<b>A</b> +037 +067	-013 +038 +069	-010 +039 +070	-008 +040 +071	-007 +041 +072	-006 +042 +073	-004 +043 +076	-003 +044	-001 +049	+014 +051	+016 +054	0RGHU +017 +055	+019 +056	+020 +057	+021 +058	+022 +059	+023 +060	+024 +061	+028 +062	+029 +063	+032 +064	+035 +065
A +037 +067 B	-013 +038 +069 -357	-010 +039 +070 +161	-008 +040 +071 +162	-007 +041 +072 +163	-006 +042 +073 -356	-004 +043 +076 -355	-003 +044 -352	-001 +049 -351	+014 +051 -350	+016 +054 -347	+017 +055 -344	M +019 +056 +340	+020 +057 +342	+021 +058 +343	+022 +059 -338	+023 +060 -337	+024 +061	+028 +062	+029 +063	+032 +064	+035 +065
A +037 +067 B -402	-013 +038 +069 -357 -399	-010 +039 +070 +161 -398	-008 +040 +071 +162 -397]	-007 +041 +072 +163 b+395	-006 +042 +073 -356 +396	-004 +043 +076 -355 +392	-003 +044 -352 +393	-001 +049 -351 +394	+014 +051 -350	\$ +016 +054 -347 -336	+017 +055 -344 -334	H +019 +056 +340 -332	+020 +057 +342 -330	+021 +058 +343 -329	+022 +059 -338 -328	+023 +060 -337 -327	+024 +061	+028 +062 -410	+029 +063 -409	+032 +064	+035 +065 -403
A +037 +067 B -402 C	-013 +038 +069 -357 -399 +202	-010 +039 +070 +161 -398 +203	-008 +040 +071 +162 -397 +200	-007 +041 +072 +163 +395 +201	-006 +042 +073 -356 +396 -198	-004 +043 +076 -355 +392 -195	-003 +044 -352 +393 -194	-001 +049 -351 +394 -193	+014 +051 -350 -388 +191	+016 +054 -347 -336 +192	+017 +055 -344 -334 +184	+019 +056 +340 -332 +185	+020 +057 +342 -330 +187	+021 +058 +343 -329 +188	+022 +059 -338 -328 -183	+023 +060 -337 -327 -182	+024 +061 -413 -181	+028 +062 -410 -179	+029 +063 -409 -177	+032 +064 -405 -175	+035 +065 -403
A +037 +067 B -402 C -425	-013 +038 +069 -357 -399 +202 -423	-010 +039 +070 +161 -398 +203 +052b	-008 +040 +071 +162 -397 +200 +053	-007 +041 +072 +163 +395 +201 +109	-006 +042 +073 -356 +396 -198 +110	-004 +043 +076 -355 +392 -195 +386	-003 +044 -352 +393 -194 +387	-001 +049 -351 +394 -193 +420	+014 +051 -350 -388 +191 -418	+016 +054 -347 -336 +192 -416	ORGHU +017 +055 -344 -334 +184 -429	M +019 +056 +340 -332 +185 -427	+020 +057 +342 -330 +187 -172	+021 +058 +343 -329 +188 -170-	+022 +059 -338 -328 -183 -169b	+023 +060 -337 -327 -182 -165	+024 +061 -413 -181 -164	+028 +062 -410 -179 +156	+029 +063 -409 -177 +157	+032 +064 -405 -175 +158	+035 +065 -403 -426 +159
A +037 +067 B -402 C -425 +160	-013 +038 +069 -357 -399 +202 -423 +154	-010 +039 +070 +161 -398 +203 +052b +155	-008 +040 +071 +162 -397 +200 +053 -153	-007 +041 +072 +163 <b>0+395</b> +201 +109 +150	-006 +042 +073 -356 +396 -198 +110 +151	-004 +043 +076 -355 +392 -195 +386 +152	-003 +044 -352 +393 -194 +387 +148	-001 +049 -351 +394 -193 +420 +149	+014 +051 -350 -388 +191 -418 -147	\$ +016 +054 -347 -336 +192 -416 -145	+017 +055 -344 -334 +184 -429 -144	M +019 +056 +340 -332 +185 -427 -137	+020 +057 +342 -330 +187 -172 +135	+021 +058 +343 -329 +188 -170 +136	+022 +059 -338 -328 -183 -169b -133	+023 +060 -337 -327 -182 -165 -132	+024 +061 -413 -181 -164 -131	+028 +062 -410 -179 +156	+029 +063 -409 -177 +157	+032 +064 -405 -175 +158	+035 +065 -403 -426 +159
A +037 +067 B -402 C -425 +160 D	-013 +038 +069 -357 -399 +202 -423 +154 -238	-010 +039 +070 +161 -398 +203 +052b +155 -237	-008 +040 +071 +162 -397 +200 +053 -153 -235	-007 +041 +072 +163 <b>b+395</b> +201 +109 +150 -234	-006 +042 +073 -356 +396 -198 +110 +151 -230	-004 +043 +076 -355 +392 -195 +386 +152 -229	-003 +044 -352 +393 -194 +387 +148 -226	-001 +049 -351 +394 -193 +420 +149 -224	+014 +051 -350 -388 +191 -418 -147 +222	s +016 +054 -347 -336 +192 -416 -145 +223	+017 +055 -344 -334 +184 -429 -144 -220	M +019 +056 +340 -332 +185 -427 -137 -218	+020 +057 +342 -330 +187 -172 +135 -217	+021 +058 +343 -329 +188 -170 +136 +210	+022 +059 -338 -328 -183 -169b -133 +211	+023 +060 -337 -327 -182 -165 -132 +212	+024 +061 -413 -181 -164 -131 +214	+028 +062 -410 -179 +156 -207	+029 +063 -409 -177 +157 -206	+032 +064 -405 -175 +158	+035 +065 -403 -426 +159
A +037 +067 B -402 C -425 +160 D E	-013 +038 +069 -357 -399 +202 -423 +154 -238 -477b	-010 +039 +070 +161 -398 +203 +052b +155 -237 -474	-008 +040 +071 +162 -397 +200 +053 -153 -235 -470	-007 +041 +072 +163 <b>b+395</b> +201 +109 +150 -234 -466b	-006 +042 +073 -356 +396 -198 +110 +151 -230 -464	-004 +043 +076 -355 +392 -195 +386 +152 -229 -460	-003 +044 -352 +393 -194 +387 +148 -226	-001 +049 -351 +394 -193 +420 +149 -224	+014 +051 -350 -388 +191 -418 -147 +222	s +016 +054 -347 -336 +192 -416 -145 +223	+017 +055 -344 -334 +184 -429 -144 -220	M +019 +056 +340 -332 +185 -427 -137 -218	+020 +057 +342 -330 +187 -172 +135 -217	+021 +058 +343 -329 +188 -170 +136 +210	+022 +059 -338 -328 -183 -169b -133 +211	+023 +060 -337 -327 -182 -165 -132 +212	+024 +061 -413 -181 -164 -131 +214	+028 +062 -410 -179 +156 -207	+029 +063 -409 -177 +157 -206	+032 +064 -405 -175 +158	+035 +065 -403 -426 +159
A +037 +067 B -402 C -425 +160 D E F	-013 +038 +069 -357 -399 +202 -423 +154 -238 -477b +128	-010 +039 +070 +161 -398 +203 +052b +155 -237 -474 +129	-008 +040 +071 +162 -397 +200 +053 -153 -235 -470 +126	-007 +041 +072 +163 <b>b+395</b> +201 +109 +150 -234 +66b +127	-006 +042 +073 -356 +396 -198 +110 +151 -230 -464 +124	-004 +043 +076 -355 +392 -195 +386 +152 -229 -460 +125	-003 +044 -352 +393 -194 +387 +148 -226 -123	-001 +049 -351 +394 -193 +420 +149 -224 -122	+014 +051 -350 -388 +191 -418 -147 +222 -121	S +016 +054 -347 -336 +192 -416 -145 +223 +111	+017 +055 -344 +184 +184 -220 +112	M +019 +056 -332 +185 -427 -137 -218 +114	+020 +057 +342 -330 +187 -172 +135 -217 +115	+021 +058 +343 -329 +188 -170 +136 +210 +117	+022 +059 -338 -328 -183 -169b -133 +211 +118	+023 +060 -337 -327 -182 -165 -132 +212 +119	+024 +061 -413 -181 -164 -131 +214 +120	+028 +062 -410 -179 +156 -207 +102	+029 +063 -409 -177 +157 -206 +104	+032 +064 -405 -175 +158 +105	+035 +065 -403 -426 +159 +106
A +037 +067 B -402 C -425 +160 D E F F -096	-013 +038 +069 -357 -399 +202 -423 +154 -238 -477b +128 +094	-010 +039 +070 +161 -398 +203 +052b +155 -237 -474 +129 +095b	-008 +040 +071 +162 -397 +200 +053 -153 -235 -470 +126 -093	-007 +041 +072 +163 <b>&gt;+395</b> +201 +109 +150 -234 466b +127 -092	-006 +042 +073 -356 +396 -198 +110 +151 -230 -464 +124 -090	-004 +043 +076 -355 +392 -195 +386 +152 -229 -460 +125 +088	-003 +044 -352 +393 -194 +387 +148 -226 -123 +089	-001 +049 -351 +394 -193 +420 +149 -224 -122 +086	+014 +051 -350 -388 +191 -418 -147 +222 -121 +087	S +016 +054 -347 -336 +192 -416 -145 +223 +111 +084	ORGHU +017 +055 -344 +184 +184 -220 +112 +085	M +019 +056 +340 -332 +185 -427 -137 -218 +114 -083	+020 +057 +342 -330 +187 -172 +135 -217 +115 -081	+021 +058 +343 -329 +188 -170 +136 +210 +117 -080	+022 +059 -338 -328 -183 -169b -133 +211 +118	+023 +060 -337 -327 -182 -132 +112 +119	+024 +061 -413 -181 -164 -131 +214 +120	+028 +062 -410 -179 +156 -207 +102	+029 +063 -409 -177 +157 -206 +104	+032 +064 -405 -175 +158 +105	+035 +065 -403 -426 +159 +106
A +037 +067 B -402 C -425 +160 D E F F -096 G	-013 +038 +069 -357 -399 +202 -423 +154 -238 +154 +128 +094 -279	-010 +039 +070 +161 -398 +203 +052b +155 -237 -474 +129 +095b -277	-008 +040 +071 +162 -397 +200 +053 -153 -235 -470 +126 -093 -276	-007 +041 +072 +163 +201 +109 +150 -234 -466b +127 -092 -274	-006 +042 +073 -356 +396 -198 +110 +151 -230 -464 +124 -090 -273	-004 +043 +076 -355 +392 -195 +386 +152 -229 -460 +125 +088 -272	-003 +044 -352 +393 -194 +387 +148 -226 -123 +089 -269	-001 +049 -351 +394 -193 +420 +149 -224 -122 +086 -268b	+014 +051 -350 -388 +191 -418 -147 +222 -121 +087 -265	s +016 +054 -347 -336 +192 -416 -145 +223 +111 +084 -263	+017 +055 -344 -334 +184 -429 -144 -220 +112 +085 +252	M +019 +056 -332 +185 -427 -137 -218 +114 -083 +260	+020 +057 +342 -330 +187 -172 +135 -217 +115 -081 +261b	+021 +058 +343 -329 +188 -170 +136 +210 +117 -080 +262	+022 +059 -338 -328 -183 -169b -133 +211 +118 -251	+023 +060 -337 -327 -182 -132 +212 +212 +119 -250	+024 +061 -181 -181 -164 -131 +214 +120 -249	+028 +062 -410 -179 +156 -207 +102 +245	+029 +063 -409 -177 +157 -206 +104 b+246	+032 +064 -405 -175 +158 +105 -244	+035 +065 -403 -426 +159 +106 -243
A +037 +067 B -402 C -425 +160 D E F F -096 G -242	-013 +038 +069 -357 -399 +202 -423 +154 -238 -477b +128 +094 -279 -239 +421	-010 +039 +070 +161 -398 +203 +052b +155 -237 -474 +129 +095b -277	-008 +040 +071 +162 -397 +200 +053 -153 -235 -470 +126 -093 -276	-007 +041 +072 +163 +201 +109 +150 -234 +66b +127 -092 -274	-006 +042 +073 -356 -198 +110 +151 -230 -464 +124 -090 -273	-004 +043 +076 -355 +392 -195 +386 +152 -229 -460 +125 +088 -272	-003 +044 -352 +393 -194 +387 +148 -226 -123 +089 -269	-001 +049 -351 +394 -193 +420 +149 -224 -122 +086 -268b	+014 +051 -350 -388 +191 -418 -147 +222 -121 +087 -265	s +016 +054 -347 -336 +192 -416 -145 +223 +111 +084 -263	+017 +055 -344 -334 +184 -429 -144 -220 +112 +085 +252	M +019 +056 +340 -332 +185 -427 -137 -218 +114 -083 +260	+020 +057 +342 -330 +187 -172 +135 -217 +115 -081 +261b	+021 +058 +343 -329 +188 -170 +136 +210 +117 -080 +262	+022 +059 -338 -328 -183 -169b -133 +211 +118 -251	+023 +060 -337 -327 -182 -165 -132 +212 +119 -250	+024 +061 -413 -181 -164 -131 +214 +120 -249	+028 +062 -410 -179 +156 -207 +102 +245	+029 +063 -409 -177 +157 -206 +104 b+246	+032 +064 -405 -175 +158 +105 -244	+035 +065 -403 -426 +159 +106 -243
A +037 +067 B -402 C -425 +160 D E F F -096 G -242 H T	-013 +038 +069 -357 -399 +202 -423 +154 -238 -477b +128 +094 -279 -239 +431 +22	-010 +039 +070 +161 -398 +203 +052b +155 -237 -474 +129 +095b -277 +4322	-008 +040 +071 +162 -397 +200 +053 -153 -235 -470 +126 -093 -276 +434	-007 +041 +072 +163 <b>&gt;+395</b> +201 +109 +150 -234 +127 -092 -274	-006 +042 +073 -356 -198 +110 +151 -230 -464 +124 -090 -273 +436c	-004 +043 +076 -355 +392 -195 +386 +152 -229 -460 +125 +088 -272 +438	-003 +044 -352 +393 -194 +387 +148 -226 -123 +089 -269 +439b	-001 +049 -351 +394 -193 +420 +149 -224 -122 +086 -268b +440 212	+014 +051 -350 -388 +191 -418 -147 +222 -121 +087 -265 +4422	S +016 +054 -347 -336 +192 -416 -145 +223 +111 +084 -263 +443 215	ORGHU +017 +055 -344 -334 +184 -220 +112 +085 +252 +447 +212	M +019 +056 +340 -332 +185 -427 -137 -218 +114 -083 +260 +448	+020 +057 +342 -330 +187 -172 +135 -217 +115 -081 +261b -450 217	+021 +058 +343 -329 +188 -170 +136 +210 +117 -080 +262 -449	+022 +059 -338 -328 -183 -169b -133 +211 +118 -251 +452	+023 +060 -337 -327 -182 -165 -132 +212 +119 -250 +454	+024 +061 -413 -181 -164 -131 +214 +120 -249 +455	+028 +062 -410 -179 +156 -207 +102 +245 +456	+029 +063 -409 -177 +157 -206 +104 b+246	+032 +064 -405 -175 +158 +105 -244	+035 +065 -403 +159 +106 -243
A +037 +067 B -025 +160 D E F F -096 G G -242 H I 1293	-013 +038 +069 -357 -399 +202 -423 +154 -238 +154 -279 -239 +431 +325 +295	-010 +039 +070 +161 -398 +203 +052b +155 -237 -474 +129 +095b -277 +432 +326	-008 +040 +071 +162 -397 +200 +053 -153 -235 -470 +126 -093 -276 +434 +323	-007 +041 +072 +163 <b>&gt;+395</b> +201 +109 +150 -234 -466b +127 -092 -274 +435c	-006 +042 +073 -356 +396 -198 +110 +151 -230 -464 +124 -090 -273 +436c -322 -290	-004 +043 +076 -355 +392 -195 +386 +152 -229 -460 +125 +088 -272 +438 -321 -321	-003 +044 -352 +393 -194 +387 +148 -226 -123 +089 -269 +439b -320 -289	-001 +049 -351 +394 -193 +420 +149 -224 -122 +086 -268b +440 -318 -286	+014 +051 -350 -388 +191 -418 -147 +222 -121 +087 -265 +442 -317b -285	S +016 +054 -347 -336 +192 -416 -145 +223 +111 +084 -263 +443 -315 -284	ORGHU +017 +055 -344 -334 +184 -220 +112 +085 +252 +447 +313 -282	M +019 +056 -332 +185 -427 -137 -218 +114 -083 +260 +448 +314	+020 +057 +342 -330 +187 -172 +135 -217 +115 -081 +261b -450 -311 -280	+021 +058 +343 -329 +188 -170 +136 +210 +117 -080 +262 -449 +138-	+022 +059 -338 -328 -183 -169b -133 +211 +118 -251 +452 +139b	+023 +060 -337 -327 -182 -182 -132 +212 +119 -250 +454 +140	+024 +061 -181 -181 -164 -131 +214 +120 -249 +455 +305	+028 +062 -410 -179 +156 -207 +102 +245 +456 +307	+029 +063 -409 -177 +157 -206 +104 b+246 +308	+032 +064 -405 -175 +158 +105 -244 +310	+035 +065 -403 426 +159 +106 -243
A +037 +067 B -402 c -425 +160 D E F F -096 G G -242 H I I +242 H I	-013 +038 +069 -399 +202 -423 +154 -238 +094 +128 +094 +128 +094 +094 +279 -239 +431 +325 +295	-010 +039 +070 +161 -398 +203 +052b +155 -237 -474 +129 +095b -277 +432 +326 +299	-008 +040 +071 +162 -397 +200 +053 -153 -235 -470 +126 -093 -276 +434 +323 +291 +279	-007 +041 +072 +163 +201 +109 +150 -234 -466b +127 -092 -274 +435c +324b +292 +380	-006 +042 +073 -356 +396 -198 +110 +151 -230 -464 +124 -090 -273 +436c -322 -290	-004 +043 +076 -355 +392 -195 +386 +152 -229 -460 +125 +088 -272 +438 -321 -289	-003 +044 -352 +393 -194 +387 +148 -226 -123 +089 -269 -269 +439b -320 -288 +392	-001 +049 -351 +394 -193 +420 +149 -224 -122 +086 -268b +440 -318 -286	+014 +051 -350 -388 +191 -418 -147 +222 -121 +087 -265 +442 -317b -285 +285	S +016 +054 -347 -336 +192 -416 -145 +223 +111 +084 -263 +443 -315 -284 +221	ORGHU +017 +055 -344 -334 +184 -429 -144 -220 +112 +085 +252 +447 +313 -283	M +019 +056 -332 +185 -427 -137 -218 +114 -083 +260 +448 +314 -281	+020 +057 +342 -330 +187 -172 +135 -217 +115 -081 +261b -450 -311 -280 +370	+021 +058 +343 -329 +188 -170 +136 +210 +117 -080 +262 -449 +138	+022 +059 -338 -328 -183 -169b -133 +211 +118 -251 +452 +139b	+023 +060 -337 -327 -182 -165 -132 +212 +119 -250 +454 +140	+024 +061 -413 -181 -164 -131 +214 +120 -249 +455 +305	+028 +062 -410 -179 +156 -207 +102 +245 +245 +307	+029 +063 -409 -177 +157 -206 +104 b+246 +308	+032 +064 -405 +175 +158 +105 -244 +310	+035 +065 -403 -426 +159 +106 -243 -304

#### Min Strip Recovery

- Formulation in [Zheng, Zhu & Sankoff 2007]
- Heuristics based on Maximum Clique [Choi et al 2007]
- NP-hard even on permutations [Wang & Zhu 2010]
- Some approximation results
  - No arbitrarily good approx. [Jiang 2011]
  - Some constant factor approx. [Chen et al. 2009]

•  $O(3^k * n)$  time algorithm, k = # of genes to delete [Jiang et al. 2012]

#### Min Strip Recovery

- Formulation in [Zheng, Zhu & Sankoff 2007]
- Heuristics based on Maximum Clique [Choi et al 2007]
- NP-hard even on permutations [Wang & Zhu 2010]
- Some approximation results
  - No arbitrarily good approx. [Jiang 2011]
  - Some constant factor approx. [Chen et al. 2009]
- $O(3^k * n)$  time algorithm, k = # of genes to delete [Jiang et al. 2012]
- This paper: polynomial time on fixed alphabets

#### <u>Exact Strip Recovery</u>

- Exact Recovery = special case with 0 deletions allowed
- Easy if input strings are permutation (no duplicates)
- Otherwise, complexity = open problem [Bulteau & Weller 2019]

#### <u>Exact Strip Recovery</u>

- Exact Recovery = special case with 0 deletions allowed
- Easy if input strings are permutation (no duplicates)
- Otherwise, complexity = open problem [Bulteau & Weller 2019]
- This paper: NP-hard if genes have duplicates

NP-hardness of a generalized Exact-Strip Recovery problem

Given two strings *S*, *T* and min block size *b*, does there exist a common partition of *S* and *T* in which the size of each block is at least *b*?

Given two strings *S*, *T* and min block size *b*, does there exist a common partition of *S* and *T* in which the size of each block is at least *b*?

b = 2: the answer is yes if and only if there is a common partition with block sizes in {2, 3} Intuition: size 4 block = 2+2, size 5 block = 2+3, ...

Given two strings *S*, *T* and min block size *b*, does there exist a common partition of *S* and *T* in which the size of each block is at least *b*?

b = 2: the answer is yes if and only if there is a common partition with block sizes in  $\{2, 3\}$ 

b = 3: the answer is yes if and only if there is a common partition with block sizes in {3,4,5}

Given two strings *S*, *T* and min block size *b*, does there exist a common partition of *S* and *T* in which the size of each block is at least *b*?

b = 2: the answer is yes if and only if there is a common partition with block sizes in  $\{2, 3\}$ 

b = 3: the answer is yes if and only if there is a common partition with block sizes in {3,4,5}

b = 4: the answer is yes if and only if there is a common partition with block sizes in {4,5,6,7}

Given two strings *S*, *T* and min block size *b*, does there exist a common partition of *S* and *T* in which the size of each block is at least *b*?

#### <u>Lemma</u>

For any infinite set F of allowed block sizes, there exists a finite set F' such that the strip recovery problem with allowed block sizes F or F' are equivalent (i.e. admit same set of solutions).

Given two strings *S*, *T* and min block size *b*, does there exist a common partition of *S* and *T* in which the size of each block is at least *b*?

#### <u>Lemma</u>

For any infinite set F of allowed block sizes, there exists a finite set F' such that the strip recovery problem with allowed block sizes F or F' are equivalent (i.e. admit same set of solutions).

• We might as well **generalize to arbitrary allowed block sizes** *F*.

The Exact F-Strip Recovery problem (XSR-F)

**Input**: two strings *S*, *T* **Question**: does there exist a common partition of *S* and *T* in which the size of each block is in *F*?

Here, *F* is a **fixed** set of integers.

The Exact F-Strip Recovery problem (XSR-F)

**Input**: two strings *S*, *T* **Question**: does there exist a common partition of *S* and *T* in which the size of each block is in *F*?

Here, *F* is a **fixed** set of integers.

#### **Theorem**

If all sizes in F are a multiple of min(F), then the problem can be solved in polynomial time. For **any other** F, the problem is NP-hard, even if each symbol **occurs at most 6 times** in the input strings.

If all sizes in F are a multiple of min(F), then the problem can be solved in polynomial time.

 $F = \{2, 4, 6\} \text{ equivalent to } F = \{2\}$ a c a b a a c b a c c b a a a c a c a b

$$F = \{2, 4, 6\} \text{ equivalent to F} = \{2\}$$
  
a c a b a a c b a c  
c b a a a c a c b a c

$$F = \{2, 4, 6\} \text{ equivalent to F} = \{2\}$$
  
a c a b a a c b a c  
c b a a a c a c b a c

#### Theorem (continued) For any other *F*, the problem is NP-hard, even if each symbol occurs at most 6 times in the input strings.

(in particular, hard for  $F = \{2,3\}$  representing blocks sizes of size at least 2)

### Theorem (continued)

For **any other** *F*, the problem is NP-hard, even if each symbol **occurs at most 6 times** in the input strings.

Reduction from *Positive Cubic 1-in-3 SAT* to XSR-F with string sets.

- Given a boolean formula in which each clause has 3 **positive** variables and each variable  $x_i$  occurs exactly three times.
- Goal: assign x<sub>i</sub>'s to *true* or *false* so that for each clause, exactly one of its variables is true.

 $(x_1 \lor x_2 \lor x_3) \land (x_1 \lor x_4 \lor x_5) \land (x_1 \lor x_2 \lor x_5) \land \dots \implies S, T$ Satisfiable iff Exact partition exists for each variable  $x_i$ Define  $X_i$  and  $X'_i$  as

 $X_{i} = P_{i}^{1} P_{i}^{2} \dots P_{i}^{r} P_{i}^{*} \mathbb{L}_{a} \mathbb{M}_{a} \mathbb{R}_{a} Q_{i} \mathbb{L}_{b} \mathbb{M}_{b} \mathbb{R}_{b} S_{i} \mathbb{L}_{c} \mathbb{M}_{c} \mathbb{R}_{c} T_{i}^{*} T_{i}^{1} T_{i}^{2} \dots T_{i}^{r}$  $X_{i}' = P_{i}^{1} P_{i}^{2} \dots P_{i}^{r} P_{i}^{*} \mathbb{L}_{a} \mathbb{M}_{a}' \mathbb{R}_{a} Q_{i} \mathbb{L}_{b} \mathbb{M}_{b}' \mathbb{R}_{b} S_{i} \mathbb{L}_{c} \mathbb{M}_{c}' \mathbb{R}_{c} T_{i}^{*} T_{i}^{1} T_{i}^{2} \dots T_{i}^{r}$  $We \text{ put } \mathcal{W}_{1} = \{X_{1}, X_{1}', X_{2}, X_{2}', \dots, X_{n}, X_{n}'\}.$ 

$$X_{i} = \underline{P_{i}^{1} \ P_{i}^{2} \ \dots \ P_{i}^{r} \ P_{i}^{*}}_{i} \underline{\mathbb{L}_{a} \ \mathbb{M}_{a}} \underline{\mathbb{R}_{a} \ Q_{i}} \underline{\mathbb{L}_{b} \ \mathbb{M}_{b}}_{i} \underline{\mathbb{R}_{b} \ S_{i}} \underline{\mathbb{L}_{c} \ \mathbb{M}_{c}}_{i} \underline{\mathbb{R}_{c} \ T_{i}^{*} \ T_{i}^{1} \ T_{i}^{2} \ \dots \ T_{i}^{r}}_{i}$$
$$X_{i}' = \underline{P_{i}^{1} \ P_{i}^{2} \ \dots \ P_{i}^{r} \ P_{i}^{*} \ \mathbb{L}_{a}}_{i} \underline{\mathbb{M}_{a}' \ \mathbb{R}_{a}}_{i} \underline{\mathbb{Q}_{i} \ \mathbb{L}_{b}}_{i} \underline{\mathbb{M}_{b}' \ \mathbb{R}_{b}}_{i} \underline{S_{i} \ \mathbb{L}_{c}} \underline{\mathbb{M}_{c}' \ \mathbb{R}_{c}}_{i} \underline{T_{i}^{*} \ T_{i}^{1} \ T_{i}^{2} \ \dots \ T_{i}^{r}}_{i}$$

**Fig. 2.** Partition of  $X_i$  and  $X'_i$  corresponding to  $x_i = true$ .

 $X_{i} = \underline{P_{i}^{1}} \ \underline{P_{i}^{2}} \ \dots \ \underline{P_{i}^{r}} \ \underline{P_{i}^{*}} \ \underline{\mathbb{L}_{a}} \ \underline{\mathbb{M}_{a}} \ \underline{\mathbb{R}_{a}} \ \underline{Q_{i}} \ \underline{\mathbb{L}_{b}} \ \underline{\mathbb{M}_{b}} \ \underline{\mathbb{R}_{b}} \ \underline{S_{i}} \ \underline{\mathbb{L}_{c}} \ \underline{\mathbb{M}_{c}} \ \underline{\mathbb{R}_{c}} \ \underline{T_{i}^{*}} \ \underline{T_{i}^{1}} \ \underline{T_{i}^{2}} \ \dots \ \underline{T_{i}^{r}} \ \underline{T_{i}^{r}} \ \underline{T$ 

**Fig. 3.** Partition of  $X_i$  and  $X'_i$  corresponding to  $x_i = false$ .

for each variable  $x_i$ Define  $X_i$  and  $X'_i$  as

 $X_{i} = P_{i}^{1} P_{i}^{2} \dots P_{i}^{r} P_{i}^{*} \mathbb{L}_{a} \mathbb{M}_{a} \mathbb{R}_{a} Q_{i} \mathbb{L}_{b} \mathbb{M}_{b} \mathbb{R}_{b} S_{i} \mathbb{L}_{c} \mathbb{M}_{c} \mathbb{R}_{c} T_{i}^{*} T_{i}^{1} T_{i}^{2} \dots T_{i}^{r}$  $X_{i}' = P_{i}^{1} P_{i}^{2} \dots P_{i}^{r} P_{i}^{*} \mathbb{L}_{a} \mathbb{M}_{a}' \mathbb{R}_{a} Q_{i} \mathbb{L}_{b} \mathbb{M}_{b}' \mathbb{R}_{b} S_{i} \mathbb{L}_{c} \mathbb{M}_{c}' \mathbb{R}_{c} T_{i}^{*} T_{i}^{1} T_{i}^{2} \dots T_{i}^{r}$  $We \text{ put } \mathcal{W}_{1} = \{X_{1}, X_{1}', X_{2}, X_{2}', \dots, X_{n}, X_{n}'\}.$ 

As for  $\mathcal{W}_2$ , each of its strings has length  $\ell$  or h. First, for each variable  $x_i$ , with  $C_a, C_b, C_c$  the clauses containing  $x_i$ , add the following strings to  $\mathcal{W}_2$ :

$$- P_i^1, P_i^2, \dots, P_i^r$$
 (r strings of length  $\ell$ )  

$$- P_i^1 P_i^2 \dots P_i^r P_i^*$$
 (one string of length  $r\ell + \ell - s = h$ }  

$$- P_i^* \mathbb{L}_a$$
 (one string of length  $\ell - s + s = \ell$ )  

$$- \mathbb{R}_a Q_i, Q_i \mathbb{L}_b, \mathbb{R}_b S_i, S_i \mathbb{L}_c$$
 (four strings of length  $s + \ell - s = \ell$ )  

$$- \mathbb{R}_c T_i^*$$
 (one string of length  $s + \ell - s = \ell$ )  

$$- T_i^* T_i^1 T_i^2 \dots T_i^r$$
 (one string of length  $r\ell + \ell - s = h$ }  

$$- T_i^1, T_i^2, \dots, T_i^r$$
 (r strings of length  $\ell$ )

Then, for each clause  $C_a$ , add the following strings to  $\mathcal{W}_2$ :

 $-\mathbb{L}_{a}\mathbb{M}_{a}, \ \mathbb{M}'_{a}\mathbb{R}_{a} \qquad (\text{two strings of length } h)$   $\mathbb{M} \mathbb{P} \quad \mathbb{I} \quad \mathbb{M}' \quad \mathbb{M} \mathbb{P} \quad \mathbb{I} \quad \mathbb{M}' \qquad (\text{four strings of length } h)$ 

#### Second step: Reduction from XSR-F with string sets to XSR-F with single sequences.

Denote  $W_1 = \{A_1, ..., A_n\}$  and  $W_2 = \{B_1, ..., B_m\}$ 

Now build single strings S and T as follows:

 $S = A_1 X_1 Y_1 Z_1 A_2 X_2 Y_2 Z_2 \dots A_n X_n Y_n Z_n$  $T = B_1 Z_1 Y_1 X_1 B_2 Z_2 Y_2 X_2 \dots B_m Z_m Y_m X_m Z_{m+1} Y_{m+1} X_{m+1} \dots Z_n Y_n X_n$ 

We now show that  $W_1, W_2$  admit a common *F*-partition if and only if *S*, *T* admit a common *F*-partition.

A polynomial time algorithm for fixed alphabet and fixed F

**Input:** two strings *S*, *T* **Goal:** delete a **minimum number of characters** from *S* and *T* so that the resulting strings admit a common partition into blocks of sizes in *F*.

**Input:** two strings *S*, *T* **Goal**: delete a **minimum number of characters** from *S* and *T* so that the resulting strings admit a common partition into blocks of sizes in *F*.

#### **Theorem**

If the alphabet  $\Sigma$  is fixed and F is also fixed, then GSR-F can be **solved in polynomial time**.

**Input:** two strings *S*, *T* **Goal**: delete a **minimum number of characters** from *S* and *T* so that the resulting strings admit a common partition into blocks of sizes in *F*.

#### **Theorem**

If the alphabet  $\Sigma$  is fixed and F is also fixed, then GSR-F can be **solved in polynomial time**. Time:  $O(n^{|F||\Sigma|^{\max(F)}+3})$ 

**Input:** two strings *S*, *T* **Goal:** delete a **minimum number of characters** from *S* and *T* so that the resulting strings admit a common partition into blocks of sizes in *F*.

#### **Theorem**

If the alphabet  $\Sigma$  is fixed and F is also fixed, then GSR-F can be **solved in polynomial time**. Time:  $O(n^{|F||\Sigma|^{\max(F)}+3})$ 

For the standard  $\Sigma = \{A, C, G, T\}$  and  $F = \{2, 3\}$ , this is  $O(n^{131})$ 

If the alphabet  $\Sigma$  is fixed and F is also fixed, then GSR-F can be **solved in polynomial time**.

Time:  $O(n^{|F||\Sigma|^{\max(F)}+3})$ 

#### **Dynamic programming algorithm**

<u>Block</u> = any string over  $\Sigma$  whose length is in *F*. <u>Block count table</u> = table *C* that assigns a number to each possible block.

aaa	aab	aba	abb	baa	bab	bba	bbb
0	0	1	0	1	0	2	0

If the alphabet  $\Sigma$  is fixed and F is also fixed, then GSR-F can be **solved in polynomial time**. Time:  $O(n^{|F||\Sigma|^{\max(F)}+3})$ 

#### **Dynamic programming algorithm**

<u>Block</u> = any string over  $\Sigma$  whose length is in *F*. <u>Block count table</u> = table *C* that assigns a number to each possible block.

For a string *S* and block count table *C*,

 $D(S, C) = \min$  number of deletions to make in S so that the resulting string can be split into blocks, such that the number of each block is the same as in C (or infinity if not possible).

For a string *S* and block count table *C*,

 $D(S, C) = \min$  number of deletions to make in S so that the resulting string can be split into blocks, such that the number of each block is the same as in C (or infinity if not possible).

C.	aaa	aab	aba	abb	baa	bab	bba	bbb
<b>L</b> .	0	0	1	0	1	0	2	0

# a b b a b a a a a b a b b a

For a string *S* and block count table *C*,

 $D(S, C) = \min$  number of deletions to make in S so that the resulting string can be split into blocks, such that the number of each block is the same as in C (or infinity if not possible).

C.	aaa	aab	aba	abb	baa	bab	bba	bbb
C.	0	0	1	0	1	0	2	0

D(S, C) = 2

### The algorithm

- On input strings *S*, *T* 
  - bestSolution =  $\infty$

For each possible block count table *C* Compute D(S, C) using DP Compute D(T, C) using DP If D(S, C) + D(T, C) < bestSolution then bestSolution = D(S, C) + D(T, C) For a string *S* and block count table *C*,

 $D(S, C) = \min$  number of deletions to make in S so that the resulting string can be split into blocks, such that the number of each block is the same as in C (or infinity if not possible).

**Idea**: for any *i*, let S[1..i] be the prefix of *S* of length *i*. D(S[1..i], C) can be computed from all the D(S[1..j], C') values, where j < i. For a string *S* and block count table *C*,

 $D(S, C) = \min$  number of deletions to make in S so that the resulting string can be split into blocks, such that the number of each block is the same as in C (or infinity if not possible).

**Idea**: for any *i*, let S[1..i] be the prefix of *S* of length *i*. D(S[1..i], C) can be computed from all the D(S[1..j], C') values, where j < i.

 $D(S[1..i], C) = \min_{X,j < i} D(S[1..j], C - X) + cost(S[j + 1..n], X)$ 

C - X means reduces the number of Xs by 1 cost(S[j + 1..n], X) is the number of deletions to make the substring become X

### The algorithm

- On input strings *S*, *T* 
  - bestSolution =  $\infty$

For each possible block count table *C* Compute D(S, C) using DP Compute D(T, C) using DP If D(S, C) + D(T, C) < bestSolution then bestSolution = D(S, C) + D(T, C)

• Complexity: dominated by the number of possible block count tables, which is  $O(n^{|F||\Sigma|^{\max(F)}})$ 

Down the theory rabbit hole: NP-hardness for given *F* 

### Complexity status

F	Σ	Complexity
Fixed	Not fixed	NP-hard (unless F is trivial)
Fixed	Fixed	P (but horrible)
Not fixed	Not fixed	?
Not fixed	Fixed	?

### Complexity status

F	Σ	Complexity
Fixed	Not fixed	NP-hard (unless F is trivial)
Fixed	Fixed	P (but horrible)
Not fixed	Not fixed	NP-hard
Not fixed	Fixed	NP-hard

#### **Theorem**

If **F is part of the input**, then the Exact F-Strip recovery problem is **NP-hard**, even if given two strings on alphabet of size 4.

#### **Theorem**

If **F** is part of the input, then the Exact F-Strip recovery problem is **NP-hard**, even if given two strings on alphabet of size 4.

Reduction from 3-Partition.

• Given a set of numbers *S* and an integer *D*, partition *S* into triples that all have the same sum *D*.

#### **Theorem**

If **F** is part of the input, then the Exact F-Strip recovery problem is **NP-hard**, even if given two strings on alphabet of size 4.

Reduction from 3-Partition.

- Given a set of numbers *S* and an integer *D*, partition *S* into triples that all have the same sum *D*.
- $S = \{a_1, \dots, a_{3m}\}, D => \text{ string } A, B \text{ and allowed sizes } F$

Reduction from 3-Partition.

- Given a set of numbers *S* and an integer *D*, partition *S* into triples that all have the same sum *D*.
- $S = \{a_1, \dots, a_{3m}\}, D => \text{ string } A, B \text{ and allowed sizes } F$

$$F = \{a_1, a_2, \dots, a_{3m}\}$$

$$A = \underbrace{\underbrace{0}_{1\dots 1}}_{B \dots 1} \underbrace{\underbrace{0}_{1\dots 1}}_{S \dots S} \underbrace{1\dots 1}_{1\dots 1} \underbrace{\underbrace{0}_{1\dots 1}}_{S \dots S} \underbrace{1\dots 1}_{M \dots S} \underbrace{1\dots 1}_{D+1} \underbrace{\frac{0}_{1\dots 1}}_{B \dots S} \underbrace{1\dots 1}_{M \dots S} \underbrace{1\dots 1}_{D+1} \underbrace{\frac{0}_{1\dots 1}}_{D+1} \underbrace{1\dots 1}_{M \dots S} \underbrace{1\dots 1}_{M \dots S} \underbrace{1\dots 1}_{M \dots M} \underbrace{1\dots 1}_{M \dots M}$$

### Conclusion

- Experiments? Nope.
  - It remains to explore the potential of strip recovery to find syntenies in practice.
- Exact algorithms? Probably not.
  - Need approximation, good heuristics, ILP, ...

# THX