AMRITA VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

22BIO211: Intelligence of Biological Systems - 2

BRANCH AND BOUND ALGORITHM FOR CYCLOPEPTIDE SEQUENCING

Dr. Manjusha Nair M
Amrita School of Computing, Amritapuri

Email : manjushanair@am.amrita.edu
Contact No: 9447745519

Brute Force Cyclopeptide Sequencing

- The Brute Force approach is slow
- The brute force method does not make good use of the spectrum given
  - *It only ever considers the largest mass value from this table*
- How might we make use of the other values?

New Approach: Cyclopeptide Sequencing

- We design a faster brute force algorithm based on a different idea.
- Previous approach (Brute Force) : Checking all cyclic peptides with a given mass.
- New approach (Branch and Bound): will "grow" candidate linear peptides whose theoretical spectra are "**consistent**" with the experimental spectrum.
- Brute force algorithms that enumerate all candidate solutions but discard large subsets of hopeless candidates by using various consistency conditions are known as branch-and-bound algorithms.
  - *Each such algorithm consists of a **branching** step to increase the number of candidate solutions, followed by a **bounding** step to remove hopeless candidates.*

A Banch and Bound Algorithm for Cyclopeptide Sequencing

- Cheking for consistency
  - *Theoretical Spectrum of Tyrocidine B1*

0	97	99	113	114	128	128	147	147	163	186	227
241	242	244	260	261	262	283	291	333	340	357	388
389	390	390	405	430	430	447	485	487	503	504	518
543	544	552	575	577	584	631	632	650	651	671	672
690	691	738	745	747	770	778	779	804	818	819	835
837	875	892	892	917	932	932	933	934	965	982	989
1031	1039	1060	1061	1062	1078	1080	1081	1095	1136	1159	1175
1175	1194	1194	1208	1209	1223	1225	1322				

  - *linear peptide VKF with spectrum : {0, 99, 128, 147, 227, 275, 374})*
  - ***Inconsistent** since every mass in its spectrum is not present in Tyrocidine B1's spectrum.*

A Banch and Bound Algorithm for Cyclopeptide Sequencing

- Cheking for consistency
  - *Theoretical Spectrum of Tyrocidine B1*

0	97	99	113	114	128	128	147	147	163	186	227
241	242	244	260	261	262	283	291	333	340	357	388
389	390	390	405	430	430	447	485	487	503	504	518
543	544	552	575	577	584	631	632	650	651	671	672
690	691	738	745	747	770	778	779	804	818	819	835
837	875	892	892	917	932	932	933	934	965	982	989
1031	1039	1060	1061	1062	1078	1080	1081	1095	1136	1159	1175
1175	1194	1194	1208	1209	1223	1225	1322				

  - *the linear peptide VKY with spectrum : ({0, 99, 128, 163, 227, 291, 390})*
  - ***is consistent** since every mass in its theoretical spectrum is present in Tyrocidine B1's spectrum.*

A Branch and Bound Algorithm for Cyclopeptide Sequencing

Given an experimental spectrum *Spectrum*,

- form a collection *Peptides* of candidate linear peptides
  - *initially consisting of the empty peptide*
    - which is just an empty string (denoted "") having mass 0.
  - *At the next step, expand Peptides to contain all linear peptides of length 1.*
  - *In next step, consider all peptides of length 2*
  - *We continue this process, creating 18 new peptides of for each amino acid string in Peptides by appending every possible amino acid mass to the end of Peptides.*

A Branch and Bound Algorithm for Cyclopeptide Sequencing

Example alphabets and masses, {A:2,B:3,C:5,D:4,E:1}

Branching step without bounding

Initialize peptides = ( )

Step 1: peptides = (A,B,C,D,E)

Mass = [2], [3], [5], [4], [1]

Step 2: peptides = (AB,AC,AD,AE,AA, BA,BB,BC,BD,BE.....)

Mass = [2, 2], [2, 3], [2, 5], [2, 4], [2, 1], [3, 2], [3, 3], [3, 5], [3, 4], [3, 1].....

Step 3: peptides = (ABA,ABB,ABC,ABD,ABE,ACA,ACB,....)

... Mass = [2, 2, 2], [2, 2, 3], [2, 2, 5], [2, 2, 4], [2, 2, 1], [2, 3, 2], [2, 3, 3], [2, 3, 5], [2, 3, 4] etc....

A Branch and Bound Algorithm for Cyclopeptide Sequencing

- To prevent the number of candidate peptides from increasing exponentially, every time we expand Peptides, we trim them by keeping only those linear peptides that remain consistent with the experimental spectrum.
- We then check if any of these new linear peptides have a mass equal to Mass (Spectrum).
- If so, we circularize this peptide and check whether it provides a solution to the Cyclopeptide Sequencing Problem.
- The branching step extends each candidate peptide of length k into 18 peptides of length k + 1, and the bounding step will remove inconsistent peptides from consideration.

A Branch and Bound Algorithm for Cyclopeptide Sequencing

```
CyclopeptideSequencing(Spectrum)
  CandidatePeptides ← a set containing only the empty peptide
  FinalPeptides ← empty list of strings
  while CandidatePeptides is nonempty
    CandidatePeptides ← Expand(CandidatePeptides)
    for each peptide Peptide in CandidatePeptides
      if Mass(Peptide) = ParentMass(Spectrum)
        if Cyclospectrum(Peptide) = Spectrum and Peptide is not in FinalPeptides
          append Peptide to FinalPeptides
        remove Peptide from CandidatePeptides
      else if Peptide is not consistent with Spectrum
        remove Peptide from CandidatePeptides
  return FinalPeptides
```

A Branch and Bound Example

- Experimental Spectrum

0	97	97	99	101	103	196	198	198	200	202
295	297	299	299	301	394	396	398	400	400	497

- first expands the set Peptides into the set of all 1-mers consistent with Spectrum:

G	A	S	P	V	T	C	I	L	N	D	K	Q	E	M	H	F	R	Y	W
57	71	87	97	99	101	103	113	113	114	115	128	128	129	131	137	147	156	163	186

97	99	101	103
P	V	T	C

A Branch and Bound Example

- The algorithm next appends each of the 18 amino acid masses to each of the 1-mers.
  - *The resulting set of Peptides containing 4 · 18 = 72 peptides of length 2 is then trimmed to keep only the 10 peptides that are consistent with Spectrum:*

97-99 PV	97-101 PT	97-103 PC	99-97 VP	99-101 VT
99-103 VC	101-97 TP	101-99 TV	103-97 CP	103-99 CV

- After expansion and trimming in the next iteration, the set Peptides contains 15 consistent 3-mers:

97-99-103 PVC	**97-99-101** **PVT**	97-101-97 PTP	**97-101-99** **PTV**	97-103-99 PCV
99-97-103 **VPC**	99-97-101 VPT	**99-101-97** **VTP**	99-103-97 VCP	101-97-99 TPV
101-97-103 TPC	**101-99-97** **TVP**	103-97-101 CPT	**103-97-99** **CPV**	103-99-97 CVP

A Branch and Bound Example

- With one more iteration, the set Peptides contains ten consistent 4-mers.

97-99-103-97 PVCP	97-101-97-99 PTPV	97-101-97-103 PTPC	97-103-99-97 PCVP
99-97-101-97 VPTP	99-103-97-101 VCPT	101-97-99-103 TPVC	101-97-103-99 TPCV
	103-97-101-97 CPTP	103-99-97-101 CVPT	

- In the final iteration, we generate ten consistent 5-mers

97-99-103-97-101 PVCPT	97-101-97-99-103 PTPVC	97-101-97-103-99 PTPCV
97-103-99-97-101 PCVPT	99-97-101-97-103 VPTPC	99-103-97-101-97 VCPTP
101-97-99-103-97 TPVCP	101-97-103-99-97 TPCVP	103-97-101-97-99 CPTPV
	103-99-97-101-97 CVPTP	

- All these linear peptides correspond to the same cyclic peptide PVCPT (Total Mass = 497)

A Branch and Bound Example -Limitation

- Observe that the six 3-mers highlighted in red failed to expand into any 4-mers in the next step

97-99-103 PVC	**97-99-101** **PVT**	97-101-97 PTP	**97-101-99** **PTV**	97-103-99 PCV
99-97-103 **VPC**	99-97-101 VPT	**99-101-97** **VTP**	99-103-97 VCP	101-97-99 TPV
101-97-103 TPC	**101-99-97** **TVP**	103-97-101 CPT	**103-97-99** **CPV**	103-99-97 CVP

- This algorithm may generate some incorrect K-mers at intermediate iterations.

97-99-103-97-101 PVCPT	97-101-97-99-103 PTPVC	97-101-97-103-99 PTPCV
97-103-99-97-101 PCVPT	99-97-101-97-103 VPTPC	99-103-97-101-97 VCPTP
101-97-99-103-97 TPVCP	101-97-103-99-97 TPCVP	103-97-101-97-99 CPTPV
	103-99-97-101-97 CVPTP	

Running Time

- Brute Force CyclopeptideSequencing is exponential
- Branch and Bound CyclopeptideSequencing is much faster, but this algorithm has not been proven to be polynomial.
- Both are inefficient since
  - *Neither algorithm's running time can be bounded by a polynomial.*

Summary

- New Approach: Cyclopeptide Sequencing
- Branch and Bound Algorithm for Cyclopeptide Sequencing Problem
- Running Time