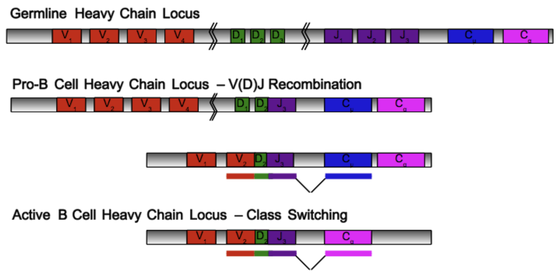
Automated Sequencing of Monoclonal Antibodies

Stefano Bonissone Natalie Castellana
Digital Proteomics LLC, La Jolla, CA

Introduction

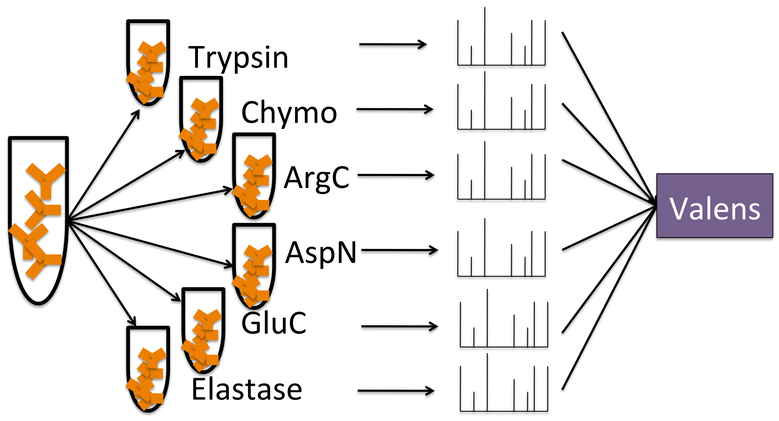
Immunoglobulins, or antibodies, are a prime example of a protein which confounds standard protein identification techniques. The mature antibody heavy chain gene results from the fusion of 4 germline genes, combinatorially chosen from a set of hundreds present in the genome. Flexible fusion boundaries, somatic hypermutation, and chemical post-translational modifications render both database search tools and de novo sequencing virtually useless.

Valens is a tool developed for sequencing monoclonal antibodies that breaks with the standard paradigm for protein identification. First, it interprets the sequence database as a template for the target protein without requiring the exact sequence to be present. By representing the sequences as a graph, Valens considers all possible V(D)J recombination events without an exponential explosion in database size. Secondly, Valens interprets multiple spectra simultaneously using the patented spectral network approach, dramatically improving identification rates.



Mass spectrometry

A single MS run (≈4500 spectra) of a single enzyme only covers about 50% of the antibody sequence. By using our recommended six enzymes (trypsin, chymotrypsin, GluC, LysC, ArgC, and ApsN), we routinely achieve over 95% coverage with Valens.



Automation

The level of automation for Valens is dataset dependent. Some datasets yield a candidate sequence with complete coverage and high spectral counts. Other datasets require additional user-intervention.

It is for this reason that we automate this process; either with a near complete candidate from the output of Valens, or from a candidate obtained from other means.

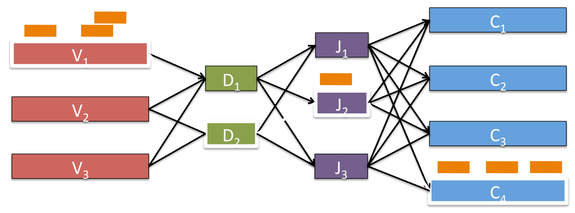
Antibody identification from mass spectra poses problems for both database search and *de novo* approaches. We integrate elements of the two basic approaches:

- Database search allows for efficient evaluation of a candidate sequence
- de novo reconstruction provides a basis for generating candidates

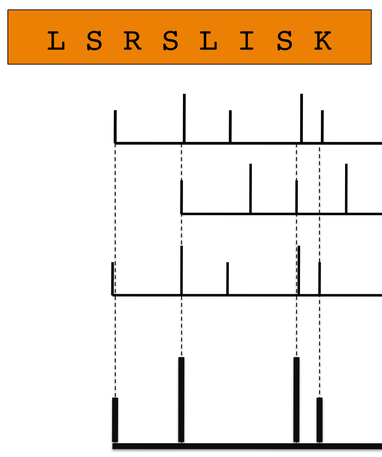
Valens Method

Stage 1: Database oriented analysis

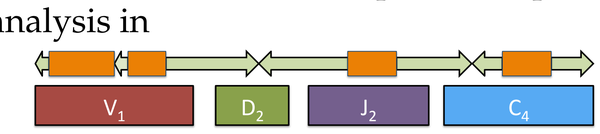
a. Select the most covered path in the template graph via database search. Identified peptides become anchors.



b. Recruit spectra that overlap anchors by spectral alignment and build a consensus



c. Repeat until anchors can be merged, or no more spectra can be recruited. Output a first prediction for analysis in



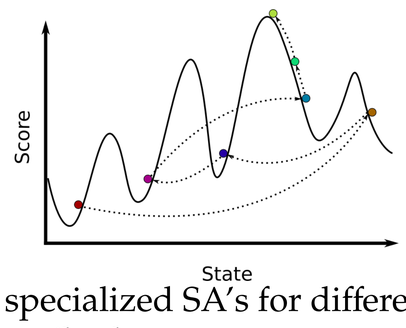
Stage 2: De novo oriented analysis

a. Recruit spectra that overlap into contigs, then *de novo* sequence each contig.

b. Align contigs to a reference protein to determine contig order and orientation.

Simulated Annealing

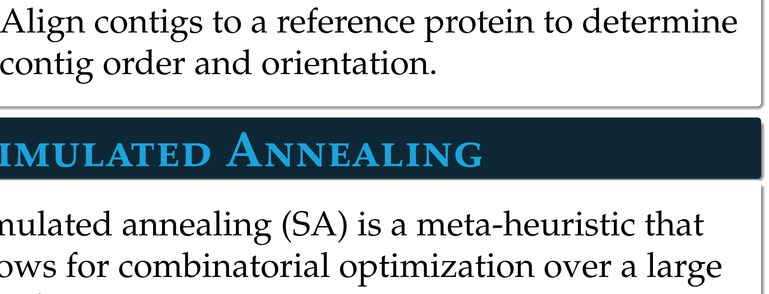
Simulated annealing (SA) is a meta-heuristic that allows for combinatorial optimization over a large search-space.



- SA's can traverse the search space by random movements
- selecting to keep poorer solutions in early iterations
- and hill-climbing in later iterations

We utilize two different specialized SA's for different regions of the antibody:

1. `SA-Variable` for uncovering mutations in the mAb variable region
   - represents the variable region as a string of amino acids
   - candidate sequence c can move to a neighbor sequence c' by substituting, inserting, or deleting a residue.
   - mutations are favored over insertions/deletions (indels)
   - substitutions are not uniform, but instead determined by a mutation matrix M.
   - M_{ij} defines the probability of mutating residue i to j, and is computed from immunoglobulin sequencing data.



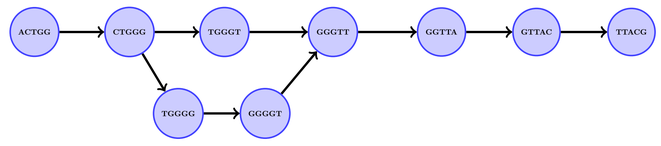
2. `SA-Junction` for closing the gap in the CDR3 junction region
   - flanking prefix/suffix sequences are taken as input
   - a seed junction sequence is used as input
   - the sequence is modified as in `SA-Variable`, but with emphasis on indels

Junction sequence

Often the junction sequence, at CDR3, creates a gap in the candidate antibody. Closing this gap is performed using a *de novo* approach by creating a de Bruijn graph.

A de Bruijn graph is generated from a set of k-mers $\mathcal{T}$, length k substrings of larger strings.

- Nodes represent all $(k-1)$-mers from $\mathcal{T}$
- An edge (u, v) is created iff u is a prefix and v is a suffix $(k-1)$-mer in some k-mer from $\mathcal{T}$



ACTGGGTTACG
ACTGGGGTTACG

Peptide tags are generated from the underlying spectra. Subsequently, a de Bruijn graph, G, is created from the k-mers of these tags. Graph G is then pruned based on the start/end of the gapped junction.

Algorithm 1: CDR3 closing algorithm

Input: S spectra, k length of k-mer, *prefix* k-mer, *suffix* k-mer

Output: P set of top N scoring sequences

1: **procedure** Close-Junction(S, k, *prefix*, *suffix*)
2: $T \leftarrow$Generate-Tags(S)
3: $G \leftarrow$Create-DeBruijn-Graph(T,k)
4: $G' \leftarrow$Prune-CDR3(G, *prefix*, *suffix*)
5: $G'' \leftarrow$Remove-cycles(G')
6: $P \leftarrow$Traverse-heaviest-paths(G'', N)
7: **end procedure**

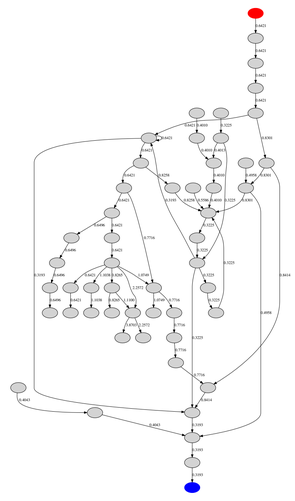
The returned assembled sequences, P, are then scored in the context of the full sequence. The top scoring candidate is used as the seed for `SA-Junc`. The use of the output of `Close-Junction` as input to the SA is two-fold:

1. Starting with a good candidate yields better results
2. `Close-Junction` alone can yield candidates

An example of the resulting graph at the junction is shown.

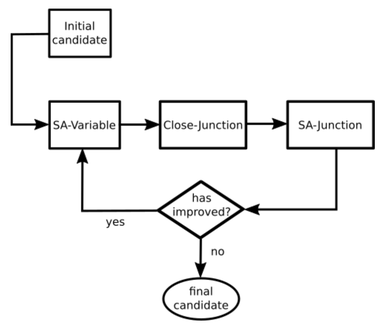
- red node denotes the start of the CDR3
- blue node denotes the end of CDR3

A graph connecting the start/end of the CDR3 is not always possible to obtain. In those instances, the best scoring paths up to a maximum depth from either end are returned.



Valens Auto

Chaining the different SA's together, we obtain the following procedure:



- The initial candidate can be obtained from Valens, or from the candidate generation shown in Step 1a.
- Stop when the candidate solution has not improved.