Table of Contents

Use Cases (Target Problems)

In developing software, we want results that will be **useful**. So, we start by asking "what does the user want to **do**?", with the aim of identifying specific problems in the domain of interest, ranging from every-day chores to complex tasks that cannot be accomplished with existing software. From studying these problems, sometimes called "use cases", we identify design criteria, then consider how to build a system. If the process works, we will end up with software that solves the problems posed by the use cases.

Here, the domain is evolutionary analysis, and our focus is not to support end-users directly (by building applications software), but to support them indirectly by providing a library. Below I will assume that the developer and the analyst are the same person, i.e., the "user" writes software to facilitate an evolutionary analysis task, then applies this software to the specific task.

Family alignment: identify homologues, generate family alignment

- **Background**: This is a core step in many analyses.
- **Key challenges**:
  - ♦ coordinate data on sequences (aa and nt)
  - ♦ allow flexibility in search and alignment
- **Preconditions**: user has a query sequence representing a protein-coding gene.
- **Steps**:
  1. user writes script to carry out automatable steps:
     a. read amino acid sequence
     b. identify homologs via database search
     c. download data on homologs
     d. compute alignment
  2. user runs script with desired parameters
  3. user visualizes output with alignment viewer or formatter

Sequence family evolution: infer sequence family tree and detect positive selection

- **Background**: a very common type of analysis, uses only sequence data and statistical models.
- **Key challenges**:
  - ♦ multiple programs needed (tree-finders do not do the dN/dS calc)
  - ♦ coordination (use tree from aa alignment, but dN/dS from nt alignment)
  - ♦ lack of developed interfaces (PAML's is lame; could make a flexible one for HyPhy)
- **Preconditions**: user has a protein sequence alignment **sequences.[aln | fas | nex | msf]** and a corresponding set of nucleotides sequences. Alternatively, the user has a coding region sequence (CDS) alignment with a genetic code for each sequence.
- **Steps**:
  1. user writes script to carry out automatable steps:
     a. read amino acid sequence alignment
     b. read corresponding codon sequences
     c. select a subset of alignment as desired
     d. infer tree from selected subset of amino acid sequence alignment
     e. compute dN/dS from selected subset of codon alignment
  2. user runs script with desired parameters
  3. user uploads output to spreadsheet for analysis or visualization

Modeling character evolution: kinase inhibitor sensitivities

- **Background**: We can obtain sequences for a large family comprising several hundred human kinases, along with inhibitor data. We wish to understand inhibitor sensitivity from a phylogenetic perspective
- **Key challenges**:
  - ♦ integrating sequence, phylogeny, structure and activity data (e.g,. name conflicts)
  - ♦ modeling IC50 evolution as a continuous character
  - ♦ visualization and data management issues due to large family size
- **Preconditions**: user has kinase sequence data; user has IC50 data for a large set of kinases in a parseable text format. User knows the binding pocket residue positions for at least one sequence.
- **Steps**:
  1. user writes script to compute alignment, get likelihood of IC50 data:
     a. read amino acid sequences
     b. compute alignment
     c. optionally read binding pocket residue positions
     d. optionally get subset of alignment corresponding to binding pocket sites
     e. infer tree from chosen (whole or subset) alignment
     f. read IC50 data
     g. re-scale IC50 data as needed
     h. compute likelihood of IC50 data given tree
  2. user runs script with desired parameters to get likelihood based on whole alignment
  3. user runs script with desired parameters to get likelihood based on binding pocket subset
  4. user uploads output to tool for visualizing ancestral states

Functional inference: identifying "functional" sites by "evolutionary trace" and related methods

- **Background**: There are several contrast methods to detect sites whose patterns of evolution change from one phylogeny-based or class-based subset to another. Such sites are candidates for specificity-determining sites where "function" has shifted during evolution. The most popular approach is "Evolutionary Trace", but the most sophisticated is the evolutionary approach of Gu.
- **Key challenges**:
  - ♦ need to develop interfaces to functional shift software
  - ♦ most methods are intended to be interactive, so an interactive interface is favorable
  - ♦ need to develop methods for visualizing results
- **Preconditions**: User has sequence data for a family of homologous proteins or protein-coding genes.
- **Steps**:
  1. user writes script to compute alignment and tree by desired methods
  2. user runs script with desired parameters to compute alignment and tree
  3. user develops graphical interface to multiple functional inference methods (CHAIN, ET (Evolutionary Trace), DIVERGE)
  4. user interactively analyses sequence family for evidence of functional shifts.

Structural genomics: identifying *Giardia* protein targets for structural characterization

- **Background**: *Giardia lamblia* is a single-celled eukaryote that can cause life-threatening instestinal disruption in humans and thus is a target for development of antimicrobial drugs. Rational drug design strategies benefit from 3D protein structures. Thus, we wish to identify the *Giardia* proteins that are most likely targets for structural characterization prior to rational drug design. The best targets will be essential to *Giardia* (so that inhibitors will be lethal), dissimilar to human proteins (so that inhibitors will not cross-react), and readily crystallized (so that the structure can be determined by x-ray

crystallography).

- **Approach**: start with the whole annotated genome; find homologs, and add data on essentiality in other organisms (worm, yeast, fly); create alignments and trees for every family; identify orthologies and paralogies; implement metrics for essentiality, distinctiveness, and crystallizability.
- **Key challenges**:
    - ♦ genome-wide analysis with thousands of different families
    - ♦ integrate external sources of data of various types

Human variation: handicapping population SNPs as potential disease alleles

- **Background**: Humans differ from each other, mainly by single-nucleotide differences called SNPs. Because SNPs are readily characterized by sequencing, they provide a basis for analyzing the significance of human genetic differences with respect to disease risks and other individual health issues (e.g., response to drugs)
- **Key challenges**:
- **Preconditions**: user has online access to SNP data and sequence databases
- **Steps**:
    1. user writes script to populate a SNP database with sequence family information
        a. for each SNP, identify genic context
        b. for SNPs in a gene
            i. get GO terms
            ii. get expression data
            iii. identify homologs (e.g., BLASTP for coding, BLASTN for non-coding)
            iv. compute alignment & phylogeny
            v. compute parameters for site-specific model
    2. user runs script, creating database
    3. user queries database
        a. sample query
        b. another query

Population analysis: to be determined

- **Background**: Weigang has suggested that we should pay more attention to this subject, given its importance. In the SNPs case above, we treat the SNPs as a collection of separate site changes that can be linked to the genome (gene, protein) sequence but have no other structure. But most population analyses introduce the complications of having multiple populations, diploidy, and linkage effects. Pedigrees and coalescents used in population analysis may be treated in some ways differently from trees.

Examples of common target problems in population analysis would be

- determining whether two population samples are from the same underlying population
- computing the coalescence time given a population sample
- determining the extent of linkage disequilibrium between two loci

Those unfamiliar with these cases might refer to the user manual for Arlequin, or look at Bio::PopGen, Bio::Variation and Bio::Pedigree::Marker::variation.

Molecular evolution: differences in data sets and methods in intron studies

- **Background**: Conflicting analyses of intron evolution lead to questions that are hard to answer. In one approach, Rogozin, et al apply Dollo parsimony to 684 KOGs families (orthologs only, one each per family from 8 complete genomes), and the results suggests a large number of introns ancestral to plants, animals and fungi. In another approach, qiu, et al apply a Bayesian time-asymmetric model to data for 10 large gene families (orthologs and paralogs, using all available data), and the results sugggest a very low probability of presence for introns in deep branches of the trees. Do the studies really conflict? If so, how much is due to different methods, and how much to different data sets?
- **Key challenges**:
    - ♦ data management issues arising from multiple data sets, methods
    - ♦ various reconstruction methods including unusual Bayesian time-asymmetric model
    - ♦ need for automated method to identify taxonomically defined nodes, e.g., PFA ancestor
- **Preconditions**: user has access to data sets.
- **Steps**:
    1. user develops database schema for reconstruction results
    2. user writes script to apply Bayesian 2-state model to KOGs data, output to database
    3. user runs script to apply Bayesian model to KOGs data and populate database with results
    4. user writes script to identify Plant-Animal-Fungal ancestor (and other relevant ancestors) in Qiu et al families
    5. user runs script to identify Plant-Animal-Fungal ancestor (and other relevant ancestors) in Qiu et al families
    6. user writes script to apply Dollo parsimony to Qiu, et al data, output to database
    7. user runs script to apply Dollo to Qiu, et al data and populate database with results
    8. user queries database to resolve differences
        a. what fraction of intron sites (expected fraction) populated with introns in PFA ancestor?
        b. how many instances (expected instances) of parallel gain?

Tree of life: whole-genome phylogeny and horizontal transfer

- **Background**:
- **Approach**:
- **Key challenges**:
    - ♦

Phylogenetic footprinting/shadowing

- **Background**: Conservation of orthologous sequences is a signal of functionality. Identification of functional genome elements based on sequence conservation is most often used for
    - ♦ Identifying gene structure (e.g., splice junctions, start codon position, ORF/exon identification)
    - ♦ Identifying non-protein coding functional elements (e.g., microRNA, promoters, and enhancers)
- **Approach**:
    1. Identify and extract orthologous sequences (through genome synteny)
    2. Align orthologous sequences
    3. Obtain a tree (maybe at the same time of alignment)
    4. Calculate total tree length of the alignment on the tree
    5. Implement the tree-length calculation in a sliding-window fashion
    6. Identify regions significantly deviates from the average tree-length
- **Challenges**:
    - ♦ Most applications have been the simplified, non-tree-based approach of using pair of sequences (e.g. Human-Mouse). Use of multiple-species comparisons, a more powerful approach, is rare.

- Alignment uncertainty.

-- WeigangQiu - 01 Sep 2006