23.1.2. Locating domains in 3D structures

(L. Holm and C. Sander)

23.1.2.1. *Introduction*

Modular design is beneficial in many areas of life, including computer programming, manufacturing, and even in protein folding.

Protein-structure analysis has long operated with the notion of domains, *i.e.*, dividing large structures into quasi-independent substructures or modules (Wetlaufer, 1973; Bork, 1992). In various contexts, these substructures are thought to fold autonomously, to carry specific molecular functions such as binding or catalysis, to move relative to each other as semi-rigid bodies and to speed the evolution of new functions by recombination (Fig. 23.1.2.1).

The problem of subdividing protein molecules into structural and functional units has received the attention of numerous researchers over the last 25 years. Early algorithms focused on protein folding or unfolding pathways and aimed at identifying substructures that would be physically stable on their own. Nowadays, with bulging macromolecular databases, the focus has shifted to devise automatic methods for identifying domains that can form the basis for a consistent protein-structure classification (Murzin *et al.*, 1995; Orengo *et al.*, 1997; Holm & Sander, 1999).

This review presents the concepts underlying computational methods for locating domains in 3D structures. Those interested in implementations are referred to the web services of the European Bioinformatics Institute* and related sites.

23.1.2.2. *Compactness*

A variety of ingenious techniques have been invented for locating structural domains in 3D structures. These include inspection of distance maps, clustering, neighbourhood correlation, plane cutting, interface area minimization, specific volume minimization, searching for mechanical hinge points, maximization of compactness and maximization of buried surface area (Rossmann & Liljas, 1974; Rashin, 1976; Crippen, 1978; Nemethy & Scheraga, 1979; Rose, 1979; Schulz & Schirmer, 1979; Go, 1981; Lesk & Rose, 1981; Sander, 1981; Wodak & Janin, 1981; Zehfus & Rose, 1986; Kikuchi *et al.*, 1988; Moult & Unger, 1991; Holm & Sander, 1994*b*; Zehfus, 1994; Islam *et al.*, 1995; Siddiqui & Barton, 1995; Swindells, 1995; Holm & Sander, 1996; Sowdhamini *et al.*, 1996; Zehfus, 1997; Holm & Sander, 1998; Jones *et al.*, 1998; Wernisch *et al.*, 1999).

Common to most approaches are the assumptions that folding units are compact and that the interactions between them are weak. These notions can be made quantitative, for example, by counting interatomic contacts and by locating domain borders by identifying groups of residues such that the number of contacts between groups is minimized. The hierarchic organization of putative folding units can be inferred starting from the complete structure and recursively cutting it (*in silico*) into smaller and smaller substructures. Alternatively, one may start from the residue or secondary-structure-element level and successively associate the most strongly interacting groups. The procedure involves two optimization problems.

The first optimization problem is algorithmic and concerns finding the optimal subdivisions. This problem is complicated by the possibility of the chain passing several times between domains (discontinuous domains). Without the constraint of sequential continuity, there is a combinatorial number of possibilities for dividing a set of residues into subsets (Zehfus, 1994). This hurdle has been overcome by fast heuristics (Holm & Sander, 1994*b*; Zehfus, 1997; Wernisch *et al.*, 1999).

The second optimization problem concerns formulating physical criteria that distinguish between autonomous and nonautonomous folding units, *i.e.*, defining termination criteria for recursive algorithms. Since compactness-related criteria do not have a clear bimodal distribution, domain-assignment algorithms (Holm & Sander, 1994*b*; Islam *et al.*, 1995; Siddiqui & Barton, 1995; Swindells, 1995; Sowdhamini *et al.*, 1996; Wernisch *et al.*, 1999) use cutoff parameters that have been fine-tuned against an external reference set of domain definitions.

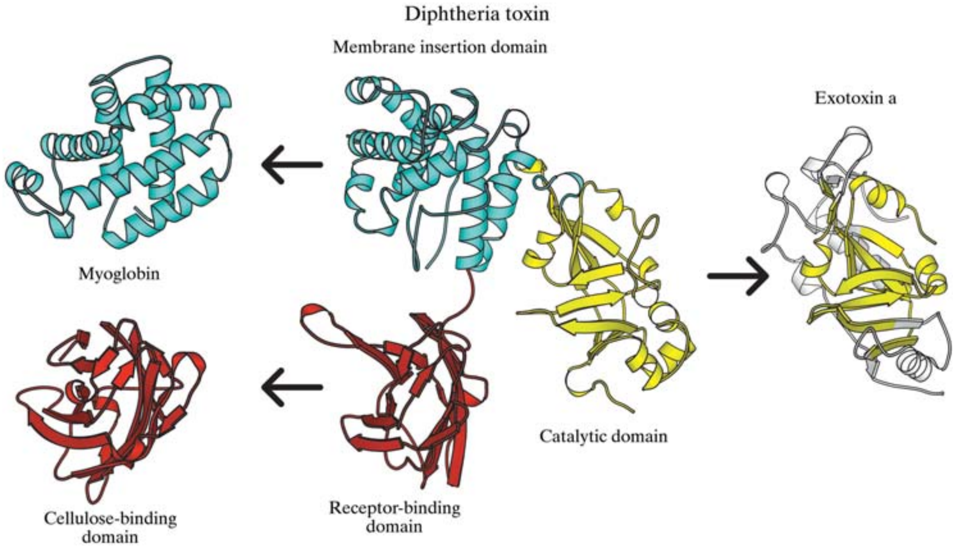

Fig. 23.1.2.1. The structure of diphtheria toxin (Bennett & Eisenberg, 1994) beautifully illustrates domains as structural, functional and evolutionary units. Structurally, note the compact globular shape of each domain and the flexible linkers between them. Functionally, note how each domain carries out a different stage of infection by the bacterium: receptor binding, membrane penetration and ADP-ribosylation of the target protein. Evolutionarily, note the occurrence of domains homologous to the catalytic domain of diphtheria toxin in exo-, entero- and pertussis toxins, and in poly-ADP-ribose polymerase (Holm & Sander, 1999). Arrows point to recurrent substructures in structural neighbours (Lionetti *et al.*, 1991; Li *et al.*, 1996; Tormo *et al.*, 1996) of each domain of diphtheria toxin. Drawn using *MOLSCRIPT* version 2 (Kraulis, 1991).

23.1.2.3. *Recurrence*

Most fold classifications use a hierarchical model where evolutionary families are a subcategory of fold type and it is natural to assume that domain boundaries should be conserved in evolution. Consistency concerns lead to a reformulation of the goals of the domain-assignment problem, away from (imprecise) physical models of stable folding units and towards recognizing such units phenomenologically in the database of known structures through recurrence. The concept of recurrence has long been the cornerstone of domain assignments by experts based on visual inspection (Richardson, 1981). Recurrence means recognizing architectural units in one protein that have already been defined (named) in another.

The practical importance of domain identification is illustrated by the discov-

* EMBL–EBI (1995): http://www.ebi.ac.uk/; DALI domain dictionary (1999): http://www.ebi.ac.uk/dali/domain/; 3Dee – database of protein domain definitions (1997): http://barton.ebi.ac.uk/servers/3Dee.html.