Proc. Natl. Acad. Sci. USA
Vol. 91, pp. 8651–8654, August 1994
Neurobiology

Overlapping neural networks for multiple motor engrams

(population vector/movement trajectory/motor skill)

ALEXANDER V. LUKASHIN*†, GEORGE L. WILCOX‡, AND APOSTOLOS P. GEORGOPOULOS*†

*Brain Sciences Center, Department of Veterans Affairs Medical Center, Minneapolis, MN 55417; †Departments of Physiology and Neurology, University of Minnesota Medical School, Minneapolis, MN 55455; and ‡Department of Pharmacology, University of Minnesota Medical School, and Minnesota Supercomputer Institute, University of Minnesota, Minneapolis, MN 55455

Communicated by Emilio Bizzi, May 31, 1994

ABSTRACT **The hypothesis was tested that learned movement trajectories of different shapes can be stored in, and generated by, largely overlapping neural networks. Indeed, it was possible to train a massively interconnected neural network to generate different shapes of internally stored, dynamically evolving movement trajectories using a general-purpose core part, common to all networks, and a special-purpose part, specific for a particular trajectory. The weights of connections between the core units do not carry any information about trajectories. The core network alone could generate externally instructed trajectories but not internally stored ones, for which both the core and the trajectory-specific part were needed. All information about the movements is stored in the weights of connections between the core part and the specialized units and between the specialized units themselves. Due to these connections the core part reveals specific dynamical behavior for a particular trajectory and, as the result, discriminates different tasks. The percentage of trajectory-specific units needed to generate a certain trajectory was small (2–5%), and the total output of the network is almost entirely provided by the core part, whereas the role of the small specialized parts is to drive the dynamical behavior. These results suggest an efficient and effective mechanism for storing learned motor patterns in, and reproducing them by, overlapping neural networks and are in accord with neurophysiological findings of trajectory-specific cells and with neurological observations of loss of specific motor skills in the presence of otherwise intact motor control.**

Although a wealth of knowledge has accumulated concerning the neural mechanisms of visually guided reaching (1–7) and tracing (8) movements, and the design and performance of artificial neural networks for similar movements (9–14), our knowledge is largely unknown concerning the generation and performance from memory of explicitly defined, learned movement trajectories, such as drawing a circle. Certain brain lesions can result in apparently specific loss of particular motor skills [“apraxia” (15, 16)], such as dressing or buttoning a garment, without affecting other motor skills (e.g., driving a car) or simple movements (e.g., reaching to a target). It is generally assumed that information concerning the performance of the lost motor skill is stored in the lesioned areas (commonly in the posterior parietal cortex) or that these areas are unique in triggering the appropriate motor action, the motor pattern of which is stored elsewhere. Whatever the mechanism, the crucial supposition is that the neural pattern of a motor skill [“motor engram” (17)] is stored somewhere *in toto* so that, when activated, it unfolds in time as a skilled motor act. Since movements are the result of interactions among neurons in the brain, it is reasonable to hypothesize that the motor engram could be stored in the set of connections and synaptic strengths between interacting neurons within and among various sensorimotor areas (18). This distributed representation of the motor skill could account for the elusiveness of the nature and the site of its motor engram (17). The neural networks subserving specific motor engrams could be separate and very specific in their composition and connection strengths, so that a particular learned, skilled action could be generated by the exclusive activation of the corresponding network with a fixed set of connection strengths (19). A generalization of this idea would posit the existence of a very large number of specific networks, one for each skill, a possible but impractical suggestion. At the other extreme, one and the same network could generate different motor behaviors by a continuous modulation of the connections in a single network (20). In this study we entertained an intermediate hypothesis—namely, that learned motor skills are subserved by largely overlapping networks with fixed connection strengths. According to this idea, the performance of a learned motor skill involves a network with two kinds of units: (*i*) general-purpose “core” units that are common to, and, therefore, engaged with, all movements and skills and (*ii*) very specialized units that are dedicated to, and, therefore, engaged with, *only* the particular set of movement trajectories comprising a motor skill. Visually guided pointing (1–7) or tracing (8) movements could be generated by the core network, whereas learned skilled movements could be generated by the concomitant activation of both the core and the specialized units. This would be a distributed mechanism by which great specificity could be achieved with a minimum of dedicated neural resources.

Model

We tested the hypothesis above by using massively interconnected neural networks modeled according to the results of experimental studies (6, 8, 21–23)—namely, (*i*) the units of the network were assigned preferred directions (6), (*ii*) the time-varying, dynamically evolving outcome of the network operation was calculated as the sum of the vectorial contribution of these units [i.e., network population vector (21, 22)], and (*iii*) such population vectors were added successively tip-to-tail to create a “neural” trajectory (8, 23). Specifically, if $\boldsymbol{C}_i$ is the unit preferred direction vector for the ith cell, then the neuronal population vector $\boldsymbol{P}$ is defined as the weighted sum of these vectors:

$$\boldsymbol{P}(t) = \sum_i V_i(t)\boldsymbol{C}_i, \quad [1]$$

where the weight $V_i(t)$ is the activity (frequency of discharge) of the ith unit at time bin t. In accordance with experimental data (6) the preferred directions were randomly and uniformly distributed in space. A neural trajectory was obtained by attaching successive population vectors:

$$\boldsymbol{R}(t_k) = \sum_{n=1}^{k} \boldsymbol{P}(t_n), \quad [2]$$

The publication costs of this article were defrayed in part by page charge payment. This article must therefore be hereby marked “*advertisement*” in accordance with 18 U.S.C. §1734 solely to indicate this fact.

where the radius-vector $\boldsymbol{R}(t_k)$ defines the point at the neural-vector trajectory taken at time bin t_k.

To calculate a neural trajectory a standard set of resistance-capacitance equations governing the interactions between units and their dynamic evolution was used (24). The time-dependent output activity of the ith unit $V_i(t)$ was calculated as $V_i(t) = \tanh[u_i(t)]$, where the variable $u_i(t)$ represents the internal state (for instance, soma membrane potential) of the unit. The dynamic evolution of the pattern of activity of N interconnected units was governed by the following system of equations:

$$\tau\frac{du_i}{dt} = -u_i(t) + \sum_{j=1}^{N} w_{ij}V_j(t) + \cos(\theta - \alpha_i), \qquad [3]$$

where $i = 1, \ldots, \mathrm{N}$; argument t is shown for values that depend on time; τ is a characteristic time constant; w_{ij} is the connection strength between units ($j \to i$). The third term on the righthand side of Eq. 3 serves to assign a preferred direction to the ith unit via extrinsic input. The angle θ corresponds to an externally given initial direction, and the angle α_i is regarded as the preferred direction of the ith unit. In the two-dimensional case the angle α_i uniquely defines the unit preferred direction vector $\boldsymbol{C}_i$. In routine calculations, Eq. 3 were solved as an initial value problem using fourth-order Runge–Kutta formula with automatic control of the step size during the integration.

The network was trained to generate four different two-dimensional complex trajectories: a clockwise circle, an orthogonal bend, a counterclockwise circle, and a sinusoid. The architecture of the neural network is illustrated in Fig. 1. Each of these trajectories was generated by a network that comprised a general-purpose core part (large ovoid) *plus* a special-purpose set of units specific for the particular trajectory (one of the small ovoids). For example, the configuration of the network illustrated in Fig. 1 implies the activation of units specific to a counterclockwise circular trajectory (double-drawn ovoid). The core part is common to, and shared by, all four networks and is, therefore, activated regardless of the shape of the trajectory: the particular shape of a trajectory depends on the specific set of units activated, together with the core units, while the remaining trajectory-specific sets are inactive. The connection strengths among the units of the core part are *fixed* and remain the same for all trajectories. On the other hand, the connection strengths between the core units and the trajectory-specific units, and those among the specific units themselves, are allowed to change during training ("*variable*" connection strengths) of the network.

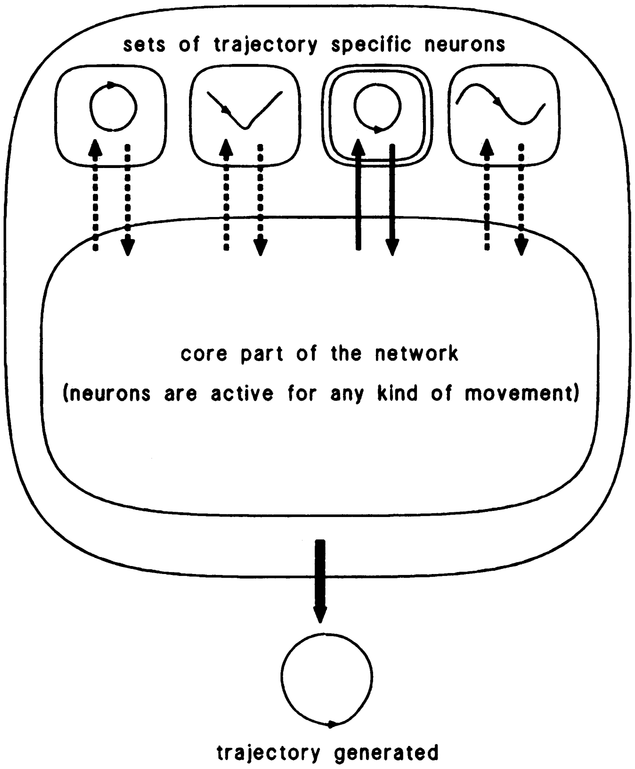

FIG. 1. Neural network for generation of four different types of neural-vector trajectories. All units in the core part are interconnected with each other and with all units in the trajectory-specific sets. All units in each trajectory-specific set are interconnected with one another and with all units in the core part. There are no connections between different trajectory-specific sets. Both kinds of heavy arrows, solid and dashed, indicate that this is a feedback network and that the connections between the core part and the trajectory-specific parts are adjusted during the training of the network to generate different types of trajectories. The double-drawn ovoid indicates that this particular set is currently activated, and connections between the core part and this set (solid arrows) determine the shape of generated neural-vector trajectory. Single-drawn ovoids indicate that the other three trajectory-specific parts of the network are inhibited and that the connections between the core part and these sets (dashed lines) do not interfere with the dynamics.

Training Procedure

To train the network to generate desired trajectories the *variable* synaptic weights were adjusted by means of the simulated annealing algorithm (25). Specifically, the simulated annealing procedure was used to minimize the root-mean-square (rms) error between the desired trajectory shape and that generated by the network:

$$F = \left(\frac{1}{K}\sum_{k=1}^{K} |\boldsymbol{R}_d(t_k) - \boldsymbol{R}_a(t_k)|^2\right)^{1/2}, \qquad [4]$$

where the radius-vectors $\boldsymbol{R}_d(t_k)$ and $\boldsymbol{R}_a(t_k)$ show the corresponding points at the desired trajectory and at actual trajectory generated by the network taken at time t_k, and $\boldsymbol{R}_a(t_0) = \boldsymbol{R}_d(t_0)$. In routine calculations we considered 200 points ($K = 200$ in Eq. 4). Each step of the simulated annealing procedure included a random change of one of the variable synaptic weights followed by an entire recalculation of the trajectory generated by the network. The new value of the synaptic weight was accepted not only for changes that lowered the rms error but also for changes that raised it. The probability of the latter event was chosen such that the system eventually obeyed the Boltzmann distribution at a given "temperature," if the rms error is treated as the "energy" of the system. The temperature was decreased according to a cooling schedule $T_{n+1} = \beta T_n$, where T_n was the temperature at the nth step and the value $1 - \beta$ was varied within the interval from $5{\cdot}10^{-4}$ to 10^{-5}. Each trial of the training procedure was repeated with different cooling schedules (different values of the parameter $1 - \beta$) to avoid the local minima problem. Generally, if the cooling is sufficiently slow for equilibrium to be established at each temperature, the global minimum—i.e., $F = 0$—can be reached in the limit of zero temperature. We checked the robustness of the results with respect to different series of random numbers used during the realization of the simulated annealing procedure. The amount of time required to train the network depended on the number of units in the simulation, on the trajectory used, and on the particular set of connection weights among the core units. For example, the computer time required on a single YMP C-90 processor ranged from 1.5 to 7.3 central processing unit hours for each trial using 100 units and the clockwise circular trajectory.

Results of Simulations and Discussion

Given the overlapping design of the networks, we sought to determine the minimal number of trajectory-specific units needed to generate a particular trajectory. Consider a core network consisting of N_C units. We first assigned and *fixed* the synaptic weights between the units of the core part. These synaptic weights were assigned randomly by the relation

$$w_{ij} = \gamma_{ij} + 0.1\left(1 - \frac{2}{\pi}\arccos(C_iC_j)\right), \qquad [5]$$

where γ_{ij} was a random number uniformly distributed on the interval $[-0.5, 0.5]$. The second term on the righthand side of Eq. 5 was introduced to provide a negative correlation between synaptic weights and the difference between preferred directions of the connected units. This type of correlation between the synaptic weights and properties of directionally tuned units was observed in experimental (5) and modeling studies (19). Then a trajectory-specific part with a given number of units ($n = 1, 2, 3, \ldots$) was added and the *variable* synaptic weights were adjusted (see above) until the network subset (i.e., core plus specific units) generated the desired trajectory, or until training failure was evident. We considered the training procedure successful if the resulting network was able to generate trajectory that provided the value of the rms error (Eq. 4) equal to or less than unity (see also Fig. 2). If this was not achieved after $3{\cdot}10^5$ steps of the simulated annealing procedure, the training was considered unsuccessful. We did not suppose any specificity of the values of variable synaptic weights in comparison with the weights of connections between the core units. During the simulated annealing procedure a new probe value for variable synaptic weight was randomly selected in accordance with Eq. 4. This means that the probe values for variable synaptic weights obeyed the same distribution function as the fixed synaptic weights for connections among the core units.

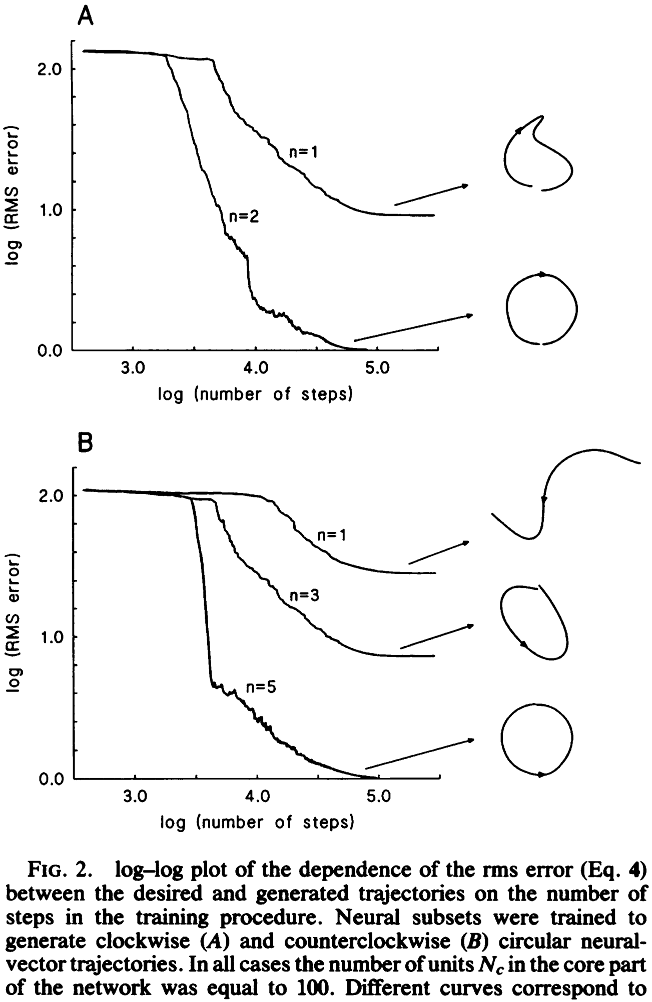

FIG. 2. log–log plot of the dependence of the rms error (Eq. 4) between the desired and generated trajectories on the number of steps in the training procedure. Neural subsets were trained to generate clockwise (*A*) and counterclockwise (*B*) circular neural-vector trajectories. In all cases the number of units N_C in the core part of the network was equal to 100. Different curves correspond to different numbers of units in the trajectory-specific parts, as indicated near the curves. The trajectories that correspond to the minimal values of the rms errors achieved in each trial are shown to the right of each curve. The training procedure was stopped if the rms error was equal to, or less than, unity. The number of specialized units n_s sufficient for this to be achieved in the trials illustrated was two and five for the clockwise (*A*) and counterclockwise (*B*) circle, respectively. The maximal number of steps in the simulated annealing was equal to $3{\cdot}10^5$.

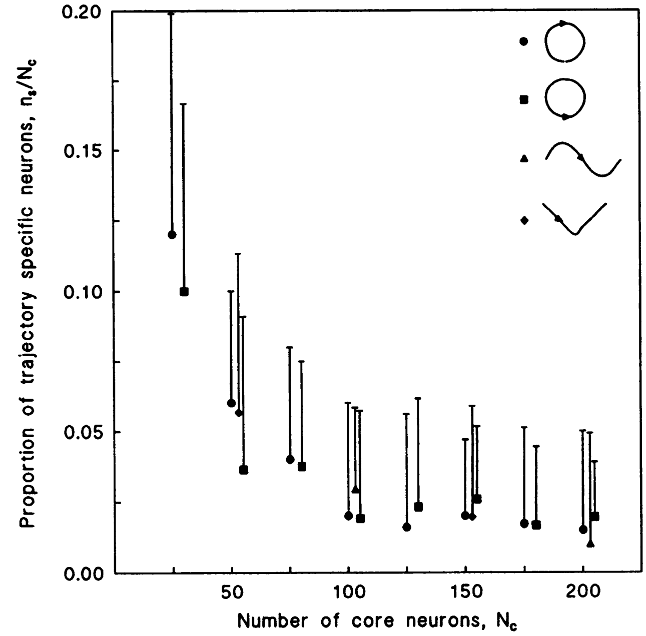

FIG. 3. The ratio of the trajectory-specific units over the number of core units, n_s/N_C, is plotted as a function of the size of the core part, N_C. The procedure of the estimation of the value n_s within one trial is described in the legend to Fig. 2. Different symbols correspond to different types of trajectories as indicated in the upper right corner. The drawn curves are examples of real trajectories generated by the subsets after the successful training. The positions of the symbols correspond to the minimal values of n_s/N_C achieved over 10 trials with the same number of the core units for different sets of synaptic weights between them. The upper edges of the vertical lines mark the maximal values of the n_s/N_C observed over the same 10 trials.

The value of n just sufficient for successful training was considered to be the minimal sufficient number of trajectory-specific units, n_s, for this trial; this number varied somewhat from trial to trial. Fig. 2 illustrates two examples of the procedure for determining the value of n_s. For a given number of units N_C in the core part, the procedure above was repeated 10 times using different but fixed synaptic weights for connections among the core units. This was carried out for each type of trajectory and for N_C values ranging from 25 to 205.

The results obtained for all simulations are shown in Fig. 3. The number of trajectory-specific units sufficient to generate a particular trajectory was small: for 100 core units, two to five trajectory-specific units were sufficient. It is noteworthy that this finding is independent of the particular shape of a trajectory (Fig. 3). Moreover, the ratio of trajectory-specific units over the number of core units (n_s/N_C) decreased as the number of core units increased. Although these results cannot be directly extrapolated to very large networks, larger simulations could yield either no improvement in accuracy (saturation) or further decreases in trajectory error. In the latter case the fraction of the trajectory-specific units, relative to the number of core units, could be even less than our estimate of 2–5%.

Thus, in the framework of our model the larger core part of the network does not carry any information about possible movements in the static state because the weights of connections between the core units are the same for all trajectories. All information about the movements is stored in the weights of connections between the core part and the specialized units and between specialized units themselves. However, once the dynamics gets started by activation of one of the specialized set, the core part reveals specific dynamical behavior for a particular trajectory, due to the driven forces from the specialized units. Therefore, during the dynamics the core part does discriminate different tasks. Note that the core part also actively influences the dynamics through the feedback connections to the specialized units. The roles played by the core part and by specialized parts are the following. Since the number of specialized units is negligibly small in comparison with the size of the core part, the total output of the whole network is almost entirely provided by the dynamical behavior of the core network that can translate the information to the lower levels of the central nervous system. The role of the small specialized parts is to receive information about the beginning of the movement and to drive the dynamical behavior of the whole network.

Recent neurophysiological studies (26–28) have shown that a small percent (1–10%) of cells recorded during performance of learned movements from memory are very specific to a particular trajectory, whereas a relatively large number of cells are engaged both during simple pointing movements *and* during performance of the specialized movements. These observations are in close quantitative agreement with the results of the present study; indeed, the experimental results can be regarded as reflecting the limit of the theoretical results obtained in this study. In the brain, specialized networks could be activated by various cortical and subcortical structures including the cerebellum and basal ganglia. Finally, the architecture of overlapping, massively interconnected networks with a minimum of specialized units could be useful to other applications requiring the production of very specific outcomes: this architecture is efficient and effective, for it maximizes the specificity that can be obtained while minimizing the number of specific units and allowing for a common core to be shared by different applications.

This research was supported by United States Public Health Service Grant PSMH48185 to A.P.G., a contract from the Office of Naval Research to A.P.G. and A.V.L., and grants of supercomputer resources from the Minnesota Supercomputer Institute to G.L.W.

1. Georgopoulos, A. P. (1990) in *Attention and Performance XIII*, ed. Jeannerod, M. (Lawrence Erlbaum, Hillsdale, AZ), pp. 227–263.
2. Georgopoulos, A. P. (1991) *Annu. Rev. Neurosci.* **14,** 361–377.
3. Kalaska, J. F. & Crammond, D. J. (1992) *Science* **255,** 1517–1523.
4. Caminiti, R. & Johnson, P. B. (1992) *Cereb. Cortex* **2,** 269–276.
5. Georgopoulos, A. P., Taira, M. & Lukashin, A. (1993) *Science* **260,** 47–52.
6. Schwartz, A. B., Kettner, R. E. & Georgopoulos, A. P. (1988) *J. Neurosci.* **8,** 2913–2927.
7. Fu, Q.-G., Suarez, J. I. & Ebner, T. J. (1993) *J. Neurophysiol.* **70,** 2097–2116.
8. Schwartz, A. B. (1993) *J. Neurophysiol.* **70,** 28–36.
9. Bullock, D. & Grossberg, S. (1988) *Psychol. Rev.* **95,** 49–90.
10. Burnod, Y., Grandguillaume, P., Otto, I., Ferraina, S., Johnson, P. B. & Caminiti, R. (1992) *J. Neurosci.* **12,** 1435–1453.
11. Kuperstein, M. (1988) *Science* **239,** 1308–1311.
12. Tononi, G., Sporns, O. & Edelman, G. M. (1992) *Cereb. Cortex* **2,** 310–335.
13. Houk, J. C. (1989) in *Models of Brain Function*, ed. Cotterill, R. M. J. (Cambridge Univ. Press, Cambridge, U.K.), pp. 309–325.
14. Lukashin, A. V. & Georgopoulos, A. P. (1993) *Biol. Cybern.* **69,** 517–524.
15. De Renzi, E. (1986) in *Diseases of the Nervous System*, eds. Asbury, A. K., McKhann, G. M. & McDonald, W. I. (Saunders, Philadelphia), pp. 848–854.
16. Poizner, H., Mack, L., Verfaellie, M., Gonzalez Rothi, L. J. & Heilman, K. M. (1990) *Brain* **113,** 85–101.
17. Lashley, K. S. (1950) *Symp. Soc. Exp. Biol.* **4,** 454–482.
18. Berthier, N. E., Singh, S. P., Barto, A. J. & Houk, J. C. (1993) *J. Cognit. Neurosci.* **5,** 56–78.
19. Lukashin, A. V. & Georgopoulos, A. P. (1994) *Neural Comput.* **6,** 19–28.
20. Harris-Warrick, R. M. & Marder, E. (1991) *Annu. Rev. Neurosci.* **14,** 39–57.
21. Georgopoulos, A. P., Caminiti, R., Kalaska, J. F. & Massey, J. T. (1983) *Exp. Brain Res. Suppl.* **7,** 327–340.
22. Georgopoulos, A. P., Schwartz, A. B. & Kettner, R. E. (1986) *Science* **233,** 1416–1419.
23. Georgopoulos, A. P., Kettner, R. E. & Schwartz, A. B. (1988) *J. Neurosci.* **8,** 2928–2937.
24. Hopfield, J. J. & Tank, D. W. (1986) *Science* **233,** 625–633.
25. Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. (1983) *Science* **220,** 671–680.
26. Hocherman, S. & Wise, S. P. (1991) *Exp. Brain Res.* **83,** 285–302.
27. Smyrnis, N., Taira, M., Ashe, J. & Georgopoulos, A. P. (1992) *Exp. Brain Res.* **92,** 139–151.
28. Ashe, J., Taira, M., Smyrnis, N., Pellizzer, G., Georgakopoulos, T., Lurito, J. T. & Georgopoulos, A. P. (1993) *Exp. Brain Res.* **95,** 118–130.