Materials Discovery 1 (2015) 21–28

Contents lists available at ScienceDirect

Materials Discovery

journal homepage: www.elsevier.com/locate/md

Automated, high throughput exploration of process–structure–property relationships using the MapReduce paradigm

Olga Wodo[a,b,c,*], Jaroslaw Zola[d,e,c], Balaji Sesha Sarath Pokuri[f], Pengfei Du[f], Baskar Ganapathysubramanian[f,g,**]

[a] University at Buffalo, Materials Design and Innovation Department, Buffalo, USA
[b] University at Buffalo, Mechanical and Aerospace Engineering Department, Buffalo, USA
[c] University at Buffalo, Computational and Data-Enabled Science and Engineering Program, Buffalo, USA
[d] University at Buffalo, Computer Science and Engineering Department, Buffalo, USA
[e] University at Buffalo, Biomedical Informatics Department, Buffalo, USA
[f] Iowa State University, Mechanical Engineering Department, Ames, USA
[g] Iowa State University, Electrical and Computer Engineering Department, Ames, USA

ARTICLE INFO

Article history:
Received 7 September 2015
Received in revised form
27 November 2015
Accepted 2 December 2015
Available online 21 December 2015

Keywords:
Process–structure–processing
High throughput analysis
MapReduce paradigm
Thermal annealing
Drift diffusion model
Organic solar cells

ABSTRACT

The microstructure of a material intimately affects the performance of a device made from this material. The microstructure, in turn, is affected by the processing pathway used to fabricate the device. This forms the process–structure–property triangle that is central to material science. There has been increasing interest to comprehensively understand and subsequently exploit process–structure–property (PSP) relationships to design processing pathways that result in tailored microstructures exhibiting optimal properties. However, unraveling process–structure–property relationships usually requires systematic and tedious combinatorial search of process and system variables to identify the microstructures that are produced. This is further complicated by the necessity to interrogate the properties of the huge set of corresponding microstructures. Motivated by this challenge, we focus on developing a generic methodology to establish and explore PSP pathways. We leverage recent advances in high performance computing (HPC) and high throughput computing (HTC) with the premise that a domain expert should be able to focus on domain specific PSP problems while the highly specialized HPC/HTC knowledge needed to approach such problems should be hidden from the domain expert. Our hypothesis is that PSP exploration can be naturally formulated in terms of a standard paradigm in cloud computing, namely the MapReduce programming model. We show how reformulating PSP exploration into a MapReduce workflow enables us to take advantage of advances in cloud computing while requiring minimal specialized knowledge of HPC. We illustrate this generic approach by exploring PSP relationships relevant to organic photovoltaics. We focus on identifying microstructural traits that correlate with specific properties of the photovoltaic process: exciton generation, exciton dissociation and charge generation. We integrate a graph-based microstructure characterization tool, and a microstructure-aware device simulator into the MapReduce workflow to automatically generate, explore and identify highly correlated microstructural traits. Identification of these microstructural traits has significant implications for designing the next generation of organic photovoltaics.

1. Introduction

Many properties of materials are microstructure-sensitive and can be modulated by carefully choosing processing conditions that lead to a tailored class of microstructures with desired properties [1–3]. The information necessary to obtain such tailored structures can be drawn from process–structure–property relationships. However, construction of process–structure–property

* Corresponding author at: University at Buffalo, Materials Design and Innovation Department, Buffalo, USA.
** Corresponding author at: Iowa State University, Mechanical Engineering Department, Ames, USA.
E-mail addresses: olgawodo@buffalo.edu (O. Wodo), jzola@buffalo.edu (J. Zola), balajip@iastate.edu (B.S. Sarath Pokuri), pdu83@iastate.edu (P. Du), baskarg@iastate.edu (B. Ganapathysubramanian).

http://dx.doi.org/10.1016/j.md.2015.12.001
2352-9245/

relationships is highly non-trivial. It requires a systematic, and invariably laborious combinatorial search of process and system variables to identify, interrogate, quantify, curate and index the (micro)structures that are produced. Consequently, the high throughput material science paradigm has been introduced to address computational challenges in large-scale materials analytics [4–11].

The idea of high throughput (combinatorial) materials science involves the (semi-) automated synthesis of a "library" of samples. The library of samples spans the quantity of interest (for example, composition, grain size, microstructure, etc.) and is usually combined with a measurement/characterization scheme to interrogate the library to identify key regions of interest, thus providing a link between processing, structure and property. This methodology has been deployed to great success by the pharmaceutical industry and has accelerated drug discovery [12–14]. It has subsequently been adopted in other areas of material science such as screening for new materials (and compositions) with desired properties including electronic, magnetic, optical properties as well as for energy-harvesting materials [6–8]. These high throughput studies have been performed using both experimental [9] as well as computational methods of analysis [10,11]. Looking at the literature available, it appears that most applications of high throughput explorations have focused on the atomistic, molecular, or composition scales [10,11] while there has been lesser focus on the microstructural scale [15].

Regardless of the application, high throughput process–structure–property construction and exploration are all characterized by a combinatorial set of possibilities and large data sets. From a computational standpoint, a fundamental challenge lies in the ability to *efficiently and automatically* explore the combinatorially large phase space of processing conditions and annotate the resulting morphologies. Consequently, three critical bottlenecks have to be addressed:

- large computational resources/power required to reliably explore the phase space *in acceptable time limits and in a fault tolerant way*,
- analyzing data in *statistically robust yet physically meaningful manner*, and
- handling the *huge and diverse data* that is produced.

Successful examples that execute a complex exploration of multi-dimensional and combinatorial space of possibilities include the search for materials for Li-ion batteries [16], molecular designs of OPV [17] or study of fluid flow behavior in micro devices [18]. All these examples rely on the availability of an *automated workflow* that orchestrates the exploration, curation and indexing process. A substantial portion of the effort is spent on designing, implementing and deploying this workflow. A critical aspect that influences the design of such workflows includes the size of the problem, both in terms of computing power required and large volume of data to be analyzed. The large problem size often mandates the use of high performance computing (HPC) tools combined with data analytics tools. The workflow must also efficiently deploy resources and maximize the exploration of the phase space with a high level of fidelity. All these software requirements suggest that design, implementation and deployment of an automated workflow for high throughput exploration requires substantial domain expertise in software engineering, fault tolerance and high performance computing. This hampers a material science domain expert from performing effective process–structure–property studies without investing substantial resources and effort in acquiring the required computational science domain expertise.

Motivated by these challenges, we explore the use of a cloud computing paradigm that automatically performs high throughput investigation in the context of establishing process–microstructure–property (PSP) relationship. This essentially hides low-level details and HPC-specific details of computational environment from the domain scientist. Our approach is based on mapping the PSP problem to a standard paradigm in Cloud Computing, namely **the MapReduce programming model**.

In recent years, the cloud has became a very attractive computational and data management resource. This is because the cloud offers flexibility, scalability, efficiency and speed in a transparent way. For example, when data is stored in the cloud, the user does not (or need not) know the physical location of the data, or even if the data is stored in one location. Data can be accessed from any geographical location in a fast and scalable way. Moreover, the cloud offers the flexibility to extend and shrink compute resources on demand. Furthermore, because of the implemented virtualization, data storage and compute requests can be easily adjusted to the specific needs. Finally, current cloud computing frameworks provide a high degree of elasticity and fault tolerance that ensure job completion, even under adverse conditions.

A cloud based framework separates the user from the low level administration of the resources. This substantially minimizes effort required from the end user to estimate, forecast and deploy storage and compute infrastructure. In most cloud platforms, the end user utilizes an easy-to-use interface that makes deployment automatic and simple. However, if desired, access to low level administrative operations is also available for optimization with respect to parallelism, data distribution, load balancing and fault tolerance. We contend that this philosophy is very attractive for domain specific experts interested in performing PSP analysis. It would be beneficial to leverage a cloud computing philosophy in material science to seamlessly exploit the increasing availability and power of HPC while maintaining the focus on material science problems. This is the main advantage of this approach compared to other workflows built for specific applications [10,16,19,20].

The focus of this paper is on illustrating how a generic PSP problem can be reformulated into the workflow under the MapReduce model to take advantage of advances in cloud computing with minimal specialized knowledge in HPC. To our best knowledge, this is the first attempt to harness cloud computing to orchestrate large scale exploration of process–structure–property space (at the microstructural level). The algorithmic details outlined in this work should serve as a template for the material science community to reformulate other high throughput materials science problems using the MapReduce paradigm. Using this generic approach, we showcase the exploring the process–structure–property relationship in organic solar cells. We specifically focus our efforts on identifying correlations between specific morphological traits and efficiency of stages of the photovoltaic process [21–24]. We identify three morphology descriptors that trace the properties of the devices. This has significant implications for the design of morphologies that result in high performance organic photovoltaic devices.

The outline of the rest of the paper is as follows: We detail the MapReduce paradigm in Section 2. In Section 3, we show how to reformulate the PSP problem as a MapReduce problem. In Section 4, we illustrate the power of this method by applying it to a complex problem involving the processing of organic photovoltaic devices. We conclude the paper in Section 5.

2. Method

We start this section with a brief overview of the MapReduce paradigm. This paradigm has been introduced to simplify parallel computing in cases where the task of interest can be expressed via two basic operations: map and reduce. The map operation transforms input prescribed by a key/value pair into a new pair. The

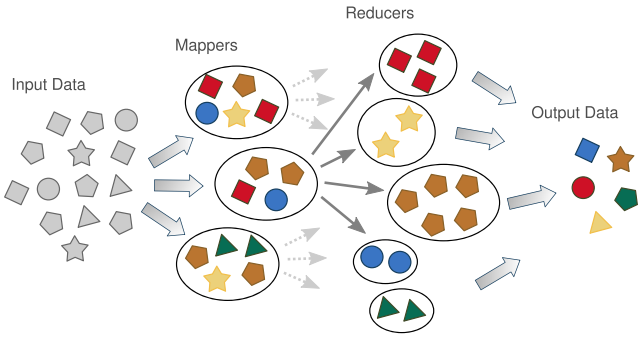

Fig. 1. Outline of MapReduce framework.

reduce operation applies user-provided reduction operator to all values with the same key.

Consider a computational framework (Process → Structure) that for a given set of processing conditions delivers a sequence of corresponding microstructures. This task can be simply formulated as a map that transforms processing variables into an annotated microstructure. Here, microstructure becomes a value and its annotation acts as a key. A straightforward annotation is some characterization of the microstructure (for example grain size distribution, orientation distribution, interfacial area, etc). Now consider a computational framework (Structure → Property) that for a given microstructure produces its set of properties. Statistically, homologous microstructures, i.e. microstructures with the same annotation, will be characterized by the same properties (Fig. 1). Hence, we can reduce microstructures into a set of equivalent property classes and then identify properties on per-class basis. In such MapReduce formulation, the only ingredients for PSP exploration are standalone (so called black box) simulators that model (a) processing effects on microstructure, (b) annotate microstructures, and (c) interrogate microstructures to compute property tags.

Once this reformulation is accomplished, the MapReduce infrastructure can be deployed to scalably, efficiently, and automatically execute the complex combinatorial task of running tens/hundreds of thousands of Process → Structure, and Structure → Property evaluations.

2.1. MapReduce paradigm

As already mentioned, MapReduce is a high level parallel programming model. Thanks to its simplicity, MapReduce has found many applications in big data analytics and machine learning [25]. It has been widely used in many disciplines, including computational biology [26,27], social networks analysis [28] and climate sciences [29], among many others. MapReduce has been implemented in many variants including for cloud environments, HPC clusters, desktop grids, volunteer computing environments and mobile clouds, which makes it a very attractive model.

The main idea behind the paradigm is to provide a restricted programming model such that programs can be automatically parallelized with transparent fault-tolerance. Specifically, a MapReduce programmer has to implement only two functions:

(i) `Map` that transforms input key/value pair into an intermediate pair that is emitted to the next stage.
(ii) `Reduce` that merges all intermediate values associated with the same intermediate key and emits the resulting key/value pair.

Once Map and Reduce functions are provided, efficient parallel and fault tolerant execution of the program is left to the underlying implementation, e.g., Hadoop [30], Spark [31], MapReduce-MPI [32], etc. The implementation takes care of scheduling, data movement, communication, and recovery in case of failure. This enables the user to focus on algorithmic components without worrying about low-level technical details. However it comes at a price of constrained flexibility. The application-specific problem must be reformulated as a series of map and reduce steps, with clearly defined key/value pairs to allow for navigation between results and orchestration of subsequent computations. Thus, the problem specific steps need to be encapsulated inside map and reduce functions. Once that is accomplished, it becomes exceedingly simple to take advantage of parallel computing including cloud computing resources. At first, this abstract way of expressing computations may appear unintuitive or confining. However, whenever a given workflow is relatively loosely coupled, it can be reformulated using MapReduce paradigm in a straightforward way. This is particularly the case for establishing Process-Structure-Property relationships where large parts of the workflow are embarrassingly parallel (independent combinatorial sweep of conditions) and can be executed as multiple, parallel and independent tasks on distributed machines.

3. Process–structure–property (PSP) exploration as MapReduce operations

A goal of PSP exploration is to build processing maps and find consistent trends that lead to: (i) improved basic understanding of underlying mechanisms of how process affects structure and how structure impacts properties, (ii) identification of promising processing conditions that produce a desired function/property, (iii) predictive capabilities guiding new (and accelerated) discovery, and finally (iv) the possibility of rapid and accurate (materials and process) design.

Establishing such links begins with a systematic exploration of processing conditions. Even the simplest manufacturing process involves at least a couple of processing parameters (e.g. for thermal annealing: annealing temperature, annealing time, the cooling rate, for forging: die speed, reduction ratio, etc.), with more complex multi-stage processes defined by several processing parameters. Thus, the 'phase space' over which the combinatorial exploration has to be performed is high-dimensional resulting in the so-called 'combinatorial explosion' of possibilities for complex processing operations. This motivates high throughput, automated, fault-tolerant materials science exploration. The MapReduce paradigm naturally facilitates this exploration.

The large scale exploration and characterization of the phase space results in a wealth of data about the final microstructure or/and evolution paths. Very often many configurations in the phase space of processing conditions result in very similar microstructure. *Consequently, it is reasonable to cluster similar structures together, and focus on representative structures*. This key observation naturally enables formulating the problem in terms of the Reduce operation. When a microstructure is emitted in the mapping stage, it is annotated with descriptor(s) that is used as a key for sorting. Subsequently, in the Reduce step, individual reducers get a list of morphologies that exhibit a similar range of descriptor (Key) values. In this way, detailed analysis of microstructures can be performed with a focus on the classes of structures, rather than on individual structures. This results in the smart use of available computational resources while covering the desired space of processing conditions with a high level of fidelity.

3.1. Expressing PSP using MapReduce

In general, the process–structure–property triangle consists of two steps: linking process with structure and linking structure with properties.

- In the first step, we sample the phase space of process variables. The phase space is sampled in an *a priori* determined manner. For every sampling point, the Process → Structure model is executed to compute the microstructure (or its evolution). Every execution is independent and can be performed as an autonomous task. Using MapReduce terminology, we say that we map the space of processing conditions and emit microstructures. In particular, we emit the microstructure (value) with its descriptor(s) (key). We use the descriptor to abbreviate/annotate the microstructure. Note that the descriptor serves as a similarity measure to cluster microstructures from different sampling points (processing conditions). The outcome of this process is a list of morphologies with their associated descriptors:
  `map(fabrication variables,-) → list(descriptor, morphology)`
- In the second step, we collect all microstructures with their associated descriptors. Microstructures with similar descriptors are considered to belong to the same class. Microstructures belonging to the same class are conceptually lumped together and denoted by a representative microstructure. Subsequently, the Structure → Property model is deployed on these representative microstructures to determine the properties of interest. Using MapReduce terminology, we say that in the reduction stage we reduce microstructures of the same descriptors and emit properties:
  `reduce (descriptor,list(morphology)) → list(properties)`

It is important to emphasize how naturally the principles of MapReduce paradigm align with the basic steps of Process–Structure–Property exploration (see Fig. 2). Thus, regardless of the application that one is interested in performing, this technique can be customized and deployed. Application specific changes can be incorporated by the user simply by defining two functions: `Map` and `Reduce`. This is illustrated in Algorithm 1, where we detail the workflow of typical execution of PSP using MapReduce paradigm.

Algorithm 1. PSP using MapReduce

Input: file with list of processing conditions, $v_0 \ldots v_n$
procedure Map key=processing condition v_i, value=empty
  `generate` config file based on given processing condition v_i
  `execute` *in silico* Process → Structure framework for sampling point v_i
  for `j=1..`n_m **do** ▷ For all n_m generated microstructures
    `execute microstructure quantification tool`
    `Emit pair(descriptor,microstructure)`
  end for
end procedure

procedure Reduce key=descriptor, value=list(microstructure)
  choose representative microstructure
  `execute` *in silico* Structure → Property framework for each representative microstructure.
  `Emit list` of properties along with *ms* and *v*
end procedure
Output: file with list of tuples: *v*, *ms*, *d*, *p* ▷ processing condition, structure, descriptor, property

We conclude this section with several remarks that we believe might be helpful when expressing specific PSP analysis via MapReduce:

- Ingredients: Three application specific standalone software packages are needed in this formulation: (a) a Process → Structure framework, (b) a Structure → Property framework, and (c) a microstructure annotation (descriptor) framework. Ingredient (a) could be any microstructure evolution framework. Representative examples include applications in organic electronics [21], polycrystal plasticity [1], etc. Ingredient (b) can be any microstructure interrogation framework. Representative examples include applications for solar cells [33], hardness [34], and thermal properties [35]. Ingredient (c) can be any method that extracts microstructure descriptors. Representative examples include graph based methods [21], FFT based methods [36], and general *n*-th order correlations [37].
- Encapsulation: The *in silico* frameworks (ingredients (a) and (b) above) can be treated as black boxes and can be swapped with different implementations as needed. Moreover, if popular MapReduce implementation, such as Hadoop, is used the burden of managing I/O can be off-loaded to this framework.
- Every microstructure is annotated with descriptor(s). This is because microstructure (most generally) is an example of unstructured data. Thus, the descriptor enables a concise and finite length representation of each arbitrary microstructure. This consistent representation serves as a key for the reduction step, where the microstructures are classified according to their descriptor values. It should be apparent that the choice of the microstructure descriptors used must be motivated by the application.
- The microstructure can be annotated with a list of descriptors. This will enable exploring the effects of several descriptors on property. Such an approach can be very valuable for identifying correlations as well as for solving inverse problem. Specifically, once the best properties are identified, such annotation facilitates the backtracking of the fabrication conditions.
- If every microstructure is of interest, then a unique key can be generated for every microstructure. In this way, no problem reduction is made and the MapReduce library runs all possible cases.
- We emphasize the simplicity of the algorithm. In practice, a few dozen lines of code is the only programming required to orchestrate the high throughput analysis to build various PSP maps. No scheduling, monitoring or rescheduling in the case of failure is required. I/O operations are handled in an automated way. For the experienced user, MapReduce libraries also deliver monitoring tools to trace the process of the execution.
- We have only illustrated one cycle of the MapReduce framework. However, multiple cycles can be easily implemented. This feature may be of importance for multistage processing. In such cases, there is a need to explore combination of various processing conditions in a nested sequence of stages. MapReduce can orchestrate such execution. Furthermore, by carefully defining reduction criteria, significant saving can be gained, as many variants in the earlier stages may lead to similar intermediate microstructures.

4. Results: PSP for organic electronics

Organic electronics (OE) covers a whole spectrum of technologies ranging from transistor, solar cells, diode lighting and flexible displays to integrated smart systems (RFIDs, smart textiles, skin). Almost from its inception, OE has ignited the imagination to build devices that exhibit flexibility, stretchability, softness, and compatibility with biological systems features traditionally missing in silicon-based solutions. OE has already revolutionized the product market of smart phones with built-in organic light emitting diode displays. However, many promising OE technologies are still bottlenecked at the property optimization stage. The underlying challenges can be traced back to a lack of understanding of the link between manufacturing and the resulting morphologies and how they impact performance. Creating designer morphologies holds the key to improved properties and their subsequent commercialization. Current progress is very often based on trial and error

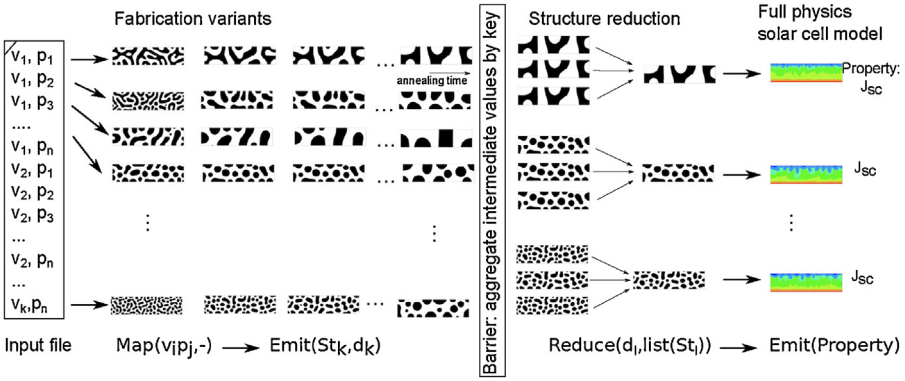

Fig. 2. Outline of the MapReduce framework developed for establishing PSP.

approaches that focus on a very narrow processing parameter space to improve properties. A fundamental challenge with such strategies – both experimental as well as computational – is to rationally explore the huge parameter space of processing conditions. Motivated by this challenge, we leverage the MapReduce paradigm to explore and identify key relationships between process, structure and property.

We focus our attention on a smaller subset of the class of OE, specifically organic solar cells (OSC). A critical research thrust over the past decade has been to enhance the power conversion efficiency of OSC. In this context, it is well known that the microstructure (or morphology) of the OSC critically affects performance. Thus, research has focused on identifying processing pathways and post-processing operation (post annealing) that can tailor the morphology to achieve higher power conversion efficiency. A critical bottleneck is that standard solution based fabrication of these thin film devices provides a large number of processing parameters that can be tuned (blend ratio, solvent type, evaporation rate, annealing temperature and time).

Therefore, our attention is on using the proposed PSP framework to identify fabrication conditions leading to an optimal morphology. We use this science question to illustrate the power of MapReduce paradigm. We perform high throughput analysis to search the phase space of different blend ratios and thermal annealing conditions. Using the results of the MapReduce paradigm we identify key morphological features that affect (correlate with) each stage of the multi-stage photovoltaic process.

4.1. Description of the individual software

In this work, we focus on the thermal annealing of the thin film consisting of a blend of two constituents (donor and acceptor). Thermal annealing is usually performed after fabrication to control the morphology evolution and obtain improved efficiency of the devices [38–40]. As stated earlier, our three *in situ* computational ingredients are:

(a) **Process → Structure framework**: the module that models morphology evolution as a function of processing conditions [41,22]. This in-house software is a modular, scalable, efficient time-adaptive finite element framework to model multi-physics (evaporation, substrate, fluid shear) driven morphology evolution in multi-component systems that describe the active layer in organic solar cells. This framework generates 2D/3D snapshots of the morphology by modeling the evolution of the morphology under the effect of processing conditions. For the results presented in this work, we deploy the framework to investigate the evolution of a binary system undergoing thermal annealing in 2D. The binary system consists of an electron donor material and an electron acceptor material. The system undergoes phase separation (due to thermal annealing). The free energy for this system is described by the Flory-Huggins free energy system with interaction parameter, χ. The domain size is 400 nm × 100 nm. The discretization used to model the system is 400 × 100. A total of 12,000 time steps were evaluated and the morphology at every 20th step is stored. We explore a range of blend ratios (concentration of donor: concentration of acceptor) and interaction parameters as our processing variables. Fig. 3 illustrates several representative morphologies that are produced as a result of these simulations.

(b) **Structure → Property framework**: the module that models the device physics of a given microstructure (more accurately, nanostructure) to compute the current–voltage operation characteristics that the particular structure produces, [33,42,43]. This in-house software is based on a finite element based solution strategy to the excitonic drift diffusion equations. The morphology-aware software solves for the spatial distribution of excitons, electrons, holes and the electric potential across the domain. The software can account for the effects of morphology by incorporating spatially varying mobilities, dielectric constants as well as interface dependent exciton dissociation, and recombination. Fig. 4 illustrates a representative result of the software for material parameters corresponding to a

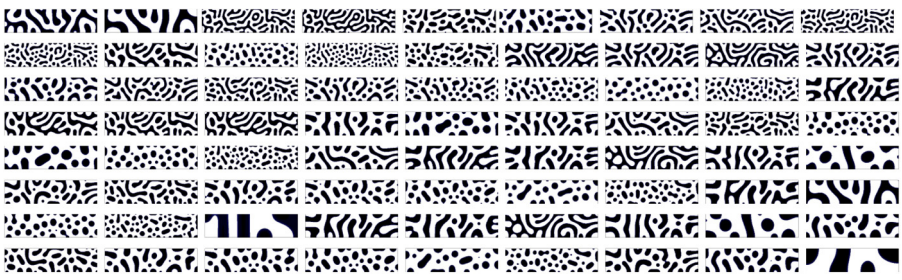

Fig. 3. Representative morphologies that are produced by the Mapper.

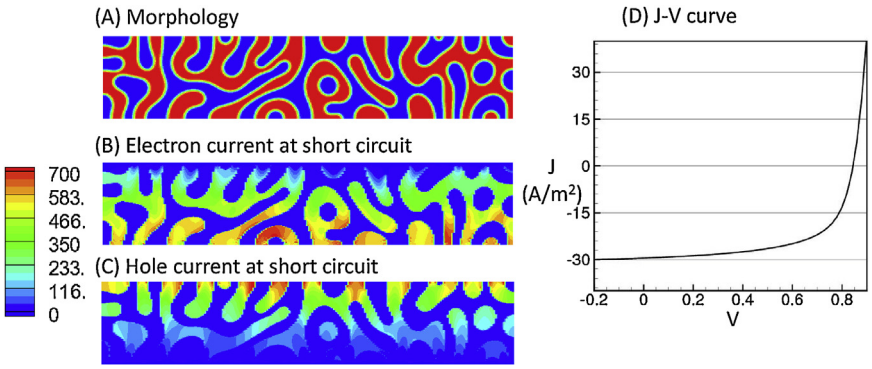

Fig. 4. Representative results of the Structure → Property framework. (A) shows a representative microstructure, (B) and (C) show the distribution of the electron and hole current densities across the domain. (D) plots the computed current–voltage curve that this particular microstructure produces.

P3HT:PCBM system [33]. In the current work we investigate the physics at short circuit (J_{sc}) conditions. Each microstructure was mapped onto a clustered mesh with a discretization of 3000×800. In addition to electron density, hole density, potential and exciton density distribution, in situ post-processing of the data was used to extract electron and hole current densities, as well as dissociation and recombination density distributions. We are particularly interested in four measures of performance: (i) the efficiency of exciton generation given as the ratio of excitons generated to incident radiation, and denoted as η_{abs}, (ii) the efficiency of exciton dissociation given as the ratio of exciton dissociated to exciton generated, and denoted as η_{diss}, (iii) the efficiency of charge transport given by the ratio of charge collected at electrodes to exciton dissociation, and denoted as η_{CT} and (iv) the short circuit current, denoted as J_{sc}.

(c) **Structure annotation framework**: the module that annotates the morphology [22]. We deploy an in-house framework that can efficiently construct a comprehensive suite of microstructure descriptors to annotate the large set of microstructures that are created. It is based on a graph-based framework to efficiently construct a broad suite of physically meaningful descriptors. These descriptors are further classified according to the physical subprocess of the photo-generation process, exciton diffusion, charge separation and charge transport. This approach is motivated by the equivalence between a discretized 2-D/3-D morphology and a labeled, weighted, undirected graph. Fig. 5 shows an example of the various descriptors that can be evaluated. A detailed discussion of this methodology is provided in [22]. We extract and annotate each morphology by a large suite of descriptors. We subsequently identify descriptors that are highly correlated with the performance measures (defined earlier). Particularly promising descriptors were: (i) fraction of domain that can absorb incident radiation (to characterize absorption), (ii) (gaussian weighted) average distance from any donor region to the donor-acceptor interface (to characterize dissociation), and (iii) average tortuosity of the domains, the fraction of the domain that was useful (percolating), the fraction of the domain with complementary paths to each electrode and contact area of preferential material with respective electrode (to characterize charge transport).

4.2. MapReduce test environment

We deployed our framework on a 31 node Hadoop cluster with a total of 248 GB main memory, and 5.4 TB secondary storage with average of 60 MB/s buffered read (as reported by hdparm-t) under the control of HDFS. The cluster has 31 nodes with dual AMD 2.2 GHz 4-core CPUs for a total of 248 cores, and uses Gigabit Ethernet for interconnect. We used a typical Hadoop configuration with one node serving as a master tracking jobs and maintaining the HDFS metadata, and remaining nodes acting as workers executing computations and storing data blocks.

The actual code required to orchestrate the entire MapReduce execution consists of less than 100 lines of code (two shell scripts). Map and reduce functions have been implemented as shell scripts that invoke the three independent SPS modules. These modules are compiled standalone applications. The complete software package constitutes a cloud-enabled framework, which we named COMA (Cloud Open Morphology Analyzer).

4.3. Correlations

We searched the phase space of blend ratio ($\phi = 0.5$–0.63) and interaction parameter ($\chi = 2.2$–4) with 10 sampling points

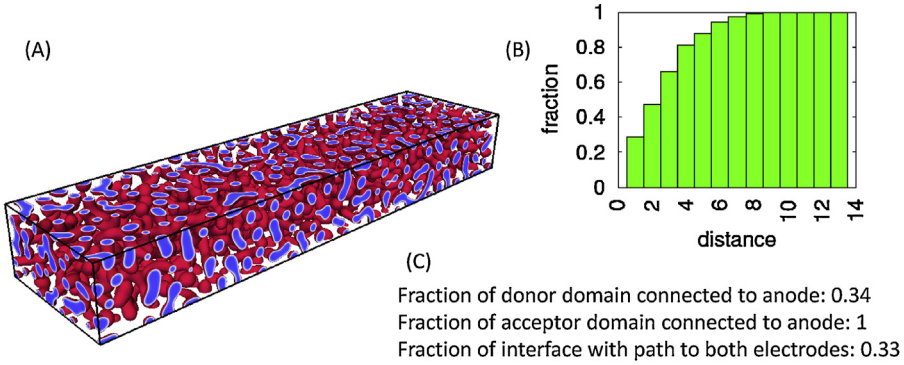

Fig. 5. Extracting morphology descriptors from a realistic 3-D morphology. (A) shows a representative morphology, (B) plots the fraction of the donor domain at a specific distance from the donor–acceptor interface. In this morphology 99% of the domain is within 8 nm to the interface. (C) shows additional morphology descriptors extracted using this framework.

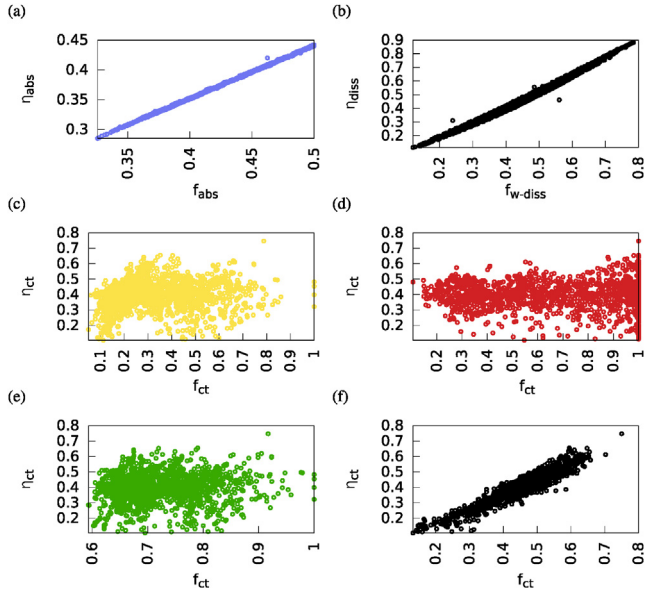

Fig. 6. Correlation study between morphology descriptors and device performance metrics: (a) correlation between the light absorption efficiency (η_{abs}) and fraction of donor material in the morphology; (b) correlation between the exciton diffusion efficiency (η_{diss}) and weighted fraction of donor material in the morphology; (c) correlation between the charge transport efficiency (η_{CT}) and the fraction of the interface with the complementary paths to both electrodes, (d) correlation between the charge transport efficiency (η_{CT}) and the fraction of domain with the straight rising paths (tortuosity $t = 1$) (e) correlation between the charge transport efficiency (η_{CT}) and the fraction of useful domains – with direct connection to the electrode (f) correlation between the charge transport efficiency (η_{CT}) and the contact area of preferential material on respective electrode.

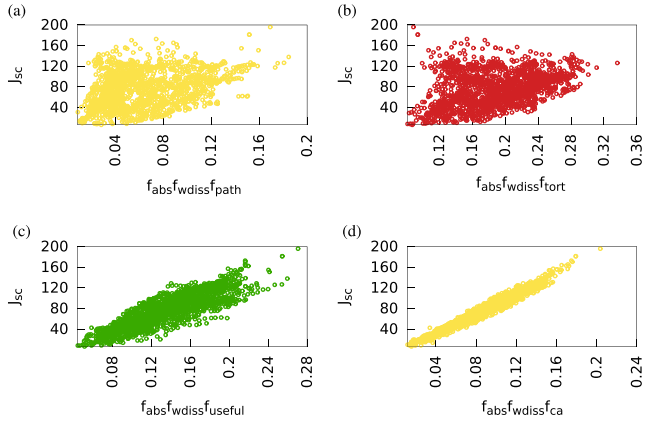

Fig. 7. Correlation study between morphology descriptors and device property (J_{sc}). Morphology descriptor constitute of three descriptors corresponding to three basic steps in the photovoltaic phenomena: light absorption, exciton diffusion and charge transport, where charge transport descriptor is changed between three options: (a) the fraction of the interface with the complementary paths to both electrodes, (b) the fraction of domain with the straight rising paths (tortuosity $t = 1$) (c), and the contact area of preferential material on respective electrode (d).

for each variables. This gives us 100 independent configurations that undergo thermal annealing. For each configuration we emit a microstructure every 20 times steps. Every microstructure is annotated with the library of descriptors as described in the previous section. Structures characterized with similar interfacial area are grouped together and two representative structures are chosen for every 100 nm^2 change in the area. We use the interfacial area as the similarity metric. The choice is application specific and is motivated by the large effect of interfacial area on the photo-physics. Specifically, the donor–acceptor interface is the only location where an exciton can charge separate. In the PSP experiment, we emitted 32,000 structures that are subsequently reduced to 2000 representative structures based on the similarity metric. Example microstructures are plotted in Fig. 3. For each microstructure in the reduced set, we run the full physics solar cell simulator and emit properties, as described in the previous section. The total execution time was ~48 h on the 32 node cluster with few failures being rescheduled by the Hadoop library.

We collect all results and ask the following science question: what is the minimal number of descriptors that comprehensively describe the photovoltaic performance of the microstructures. To answer this question, we perform correlation studies by pairing each microstructure descriptor with each full-physics metrics that was computed. Note that three of the physics based metrics encode each of the three stages of the photo-physics during OSC operation: light absorption efficiency, exciton diffusion efficiency and charge transport efficiency.

Fig. 6 plots some of the most promising descriptors. For absorption efficiency the best descriptor (with a very high correlation coefficient, ~0.99) is the volume fraction of the donor (Fig. 6(a)). This intuitively makes sense as the donor is the material that absorbs incident light to create excitons. In the case of exciton dissociation efficiency, the most promising descriptor is the weighted fraction of averaged donor domain. Fig. 6 (b) plots the correlation between this descriptor and the exciton dissociation efficiency. Note that exciton undergoes diffusive transport across the donor domain. Thus, a weighting function that reflects this random walk (a gaussian weighting function) encodes the physics of exciton diffusion and subsequent dissociation [24]. Fig. 6(c–f) show correlation plots for three different descriptors that are correlated with the charge transport efficiency. All three descriptors give less than satisfactory correlations, with the best correlation achieved for the descriptor that measures the fraction of domains with direct connection to the respective electrodes (contact area of donor domains on the anode, and contact area of accepting material on the cathode).

We next combine these descriptors to perform correlation analysis with the key quantity of interest – the short circuit current, J_{sc}. We identify the best combination(s) of microstructure descriptors that **predict** the short circuit current. We consider products of descriptors to perform correlation analysis with J_{sc}. Specifically, we consider (triple) products of descriptors that correlate with absorption (1 descriptor), dissociation (1 descriptor) and charge transport (3 descriptors), respectively. Fig. 7 plots the four possibilities. The results indicate that using the fraction of the domain directly connected to the electrode as the charge transport descriptor (along with the absorption descriptor and the dissociation descriptor) predicts well the short circuit performance of a microstructure (with a ~0.85 Pearson's linear correlation coefficient). However, the best correlation we report is for the contact area of preferential material with the preferential electrode (with a ~0.98 Pearson's linear correlation coefficient). The identification of this joint descriptor that correlates well with J_{sc} is a significant breakthrough towards rapid identification and rank ordering of morphologies in terms of their photo-voltaic performance. Currently, several morphological features are discussed and used by the community in the context of evaluating photovoltaic performance in OSC. The results reported in this paper enable a rigorous ranking of morphological features and illustrate their utility in evaluating the performance of OSC. These three morphology descriptors can be used as a basis to formulate a morphology design framework to identify morphologies that will deliver improved OSC performance. Specifically, using these descriptors we can perform quick screening of morphologies or use it as a cost function for the topological optimization of morphologies. To our best knowledge this is the first successful attempt to find the minimal number of descriptors that comprehensively

describe the photovoltaic performance of the microstructures. This analysis and result would have been extremely time-consuming to produce without the MapReduce framework utilized here.

5. Conclusions

We describe a methodology to reformulate the challenge of high throughput exploration during Process–Structure–Property analysis into the workflow under the MapReduce model in order to take advantage of advances in cloud computing with minimal specialized knowledge in HPC. We hope that the algorithmic details outlined in this work will serve as a template for the material science community to reformulate other high throughput materials science problems using the MapReduce paradigm. We showcase this generic approach in the context of exploring the process–structure–property relationships in organic solar cells. We specifically focus our efforts on identifying correlations between specific morphological traits and efficiency of stages of the photovoltaic process. Through the high throughput analysis, we found three morphological traits that correlate well with the short circuit current. This has significant implications for the design of morphologies for high performance organic photovoltaic devices.

Acknowledgment

This research has been supported in part by NSF CAREER 1149365, NSF 1435587 and NSF XSEDE.

References

[1] V. Sundararaghavan, N. Zabaras, Design of microstructure-sensitive properties in elasto-viscoplastic polycrystals using multi-scale homogenization, Int. J. Plast. 22 (10) (2006) 1799–1824.
[2] D.T. Fullwood, S.R. Niezgoda, B.L. Adams, S.R. Kalidindi, Microstructure sensitive design for performance optimization, Prog. Mater. Sci. 55 (6) (2010) 477–562.
[3] S. Ganapathysubramanian, N. Zabaras, Design across length scales: a reduced-order model of polycrystal plasticity for the control of microstructure-sensitive material properties, Comput. Methods Appl. Mech. Eng. 193 (45) (2004) 5017–5034.
[4] J.R. Nowers, S.R. Broderick, K. Rajan, B. Narasimhan, Combinatorial methods and informatics provide insight into physical properties and structure relationships during IPN formation, Macromol. Rapid Commun. 28 (8) (2007) 972–976.
[5] C. Suh, K. Rajan, Combinatorial design of semiconductor chemistry for bandgap engineering: virtual combinatorial experimentation, Appl. Surf. Sci. 223 (1) (2004) 148–158.
[6] M.L. Green, I. Takeuchi, J.R. Hattrick-Simpers, Applications of high throughput (combinatorial) methodologies to electronic, magnetic, optical, and energy-related materials, J. Appl. Phys. 113 (23) (2013) 231101.
[7] S. Curtarolo, G.L. Hart, M.B. Nardelli, N. Mingo, S. Sanvito, O. Levy, The high-throughput highway to computational materials design, Nat. Mater. 12 (3) (2013) 191–201.
[8] K. Rajan, Informatics for Materials Science and Engineering: Data-driven Discovery for Accelerated Experimentation and Application, Butterworth-Heinemann, 2013.
[9] H. Duan, C. Yuan, N. Becerra, L. Small, A. Chang, J. Gregoire, R. van Dover, High-throughput measurement of ionic conductivity in composition-spread thin films, ACS Comb. Sci. 15 (6) (2013) 273–277.
[10] A. Jain, G. Hautier, C.J. Moore, S.P. Ong, C.C. Fischer, T. Mueller, K.A. Persson, G. Ceder, A high-throughput infrastructure for density functional theory calculations, Comput. Mater. Sci. 50 (8) (2011) 2295–2310.
[11] J. Greeley, T.F. Jaramillo, J. Bonde, I. Chorkendorff, J.K. Nørskov, Computational high-throughput screening of electrocatalytic materials for hydrogen evolution, Nat. Mater. 5 (11) (2006) 909–913.
[12] N. Beeley, A. Berger, A revolution in drug discovery: Combinatorial chemistry still needs logic to drive science forward, Br. Med. J. 321 (7261) (2000) 581.
[13] R. Merrifield, Automated synthesis of peptides, Science 150 (3693) (1965) 178–185.
[14] L.A. Thompson, J.A. Ellman, Synthesis and applications of small molecule libraries, Chem. Rev. 96 (1) (1996) 555–600.
[15] S.R. Kalidindi, Data science and cyberinfrastructure: critical enablers for accelerated development of hierarchical materials, Int. Mater. Rev. 60 (3) (2015) 150–168.
[16] S.P. Ong, W.D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V.L. Chevrier, K.A. Persson, G. Ceder, Python materials genomics (pymatgen): a robust, open-source python library for materials analysis, Comput. Mater. Sci. 68 (2013) 314–319.
[17] J. Hachmann, R. Olivares-Amaya, S. Atahan-Evrenk, C. Amador-Bedolla, R.S. Sánchez-Carrera, A. Gold-Parker, L. Vogt, A.M. Brockway, A. Aspuru-Guzik, The Harvard clean energy project: large-scale computational screening and design of organic photovoltaics on the world community grid, J. Phys. Chem. Lett. 2 (17) (2011) 2241–2251.
[18] J. Diaz-Montes, Y. Xie, I. Rodero, J. Zola, B. Ganapathysubramanian, M. Parashar, Federated computing for the masses-aggregating resources to tackle large-scale engineering problems, Comput. Sci. Eng. 16 (4) (2014) 62–72.
[19] S. Curtarolo, W. Setyawan, G.L. Hart, M. Jahnatek, R.V. Chepulskii, R.H. Taylor, S. Wang, J. Xue, K. Yang, O. Levy, et al., Aflow: an automatic framework for high-throughput materials discovery, Comput. Mater. Sci. 58 (2012) 218–226.
[20] S.P. Ong, S. Cholia, A. Jain, M. Brafman, D. Gunter, G. Ceder, K.A. Persson, The materials application programming interface (api): a simple, flexible and efficient api for materials data based on representational state transfer (rest) principles, Comput. Mater. Sci. 97 (2015) 209–215.
[21] O. Wodo, S. Tirthapura, S. Chaudhary, B. Ganapathysubramanian, Computational characterization of bulk heterojunction nanomorphology, J. Appl. Phys. 112 (6) (2012) 064316.
[22] O. Wodo, S. Tirthapura, S. Chaudhary, B. Ganapathysubramanian, A graph-based formulation for computational characterization of bulk heterojunction morphology, Org. Electron. 13 (6) (2012) 1105–1113.
[23] A. Aboulhassan, D. Baum, O. Wodo, B. Ganapathysubramanian, A. Amassian, M. Hadwiger, A novel framework for visual detection and exploration of performance bottlenecks in organic photovoltaic solar cell materials, in: Computer Graphics Forum, Vol. 34, 2015, pp. 401–410.
[24] O. Wodo, J.D. Roehling, A.J. Moulé, B. Ganapathysubramanian, Quantifying organic solar cell morphology: a computational study of three-dimensional maps, Energy Environ. Sci. 6 (10) (2013) 3060–3070.
[25] J. Dean, S. Ghemawat, Mapreduce: simplified data processing on large clusters, Commun. ACM 51 (1) (2008) 107–113.
[26] E.E. Schadt, M.D. Linderman, J. Sorenson, L. Lee, G.P. Nolan, Computational solutions to large-scale data management and analysis, Nat. Rev. Genet. 11 (9) (2010) 647–657.
[27] X. Yang, J. Zola, S. Aluru, Parallel metagenomic sequence clustering via sketching and maximal quasi-clique enumeration on map-reduce clouds, in: IEEE International Parallel & Distributed Processing Symposium (IPDPS), IEEE, 2011, pp. 1223–1233.
[28] H. Yin, J. Li, Y. Niu, Detecting local communities within a large scale social network using mapreduce, Int. J. Intel. Inf. Technol.) 10 (1) (2014) 57–76.
[29] K. Ajay, K. Gouda, H. Nagesh, A study for handelling of high-performance climate data using hadoop, Int. J. Intel. Inf. Technol. (2015) 197–202.
[30] Apache, http://hadoop.apache.org/.
[31] Spark, http://spark.apache.org/.
[32] Mapreduce-mpi library, http://mapreduce.sandia.gov/.
[33] H.K. Kodali, B. Ganapathysubramanian, Computer simulation of heterogeneous polymer photovoltaic devices, Model. Simul. Mater. Sci. Eng. 20 (3) (2012) 035015.
[34] S.R. Kalidindi, C.A. Bronkhorst, L. Anand, Crystallographic texture evolution in bulk deformation processing of FCC metals, J. Mech. Phys. Solids 40 (3) (1992) 537–569.
[35] B. Ganapathysubramanian, N. Zabaras, Modeling diffusion in random heterogeneous media: data-driven models, stochastic collocation and the variational multiscale method, J. Comput. Phys. 226 (1) (2007) 326–353.
[36] S.R. Niezgoda, D.M. Turner, D.T. Fullwood, S.R. Kalidindi, Optimized structure based representative volume element sets reflecting the ensemble-averaged 2-point statistics, Acta Mater. 58 (13) (2010) 4432–4445.
[37] S. Torquato, Random Heterogeneous Materials: Microstructure and Macroscopic Properties, Vol. 16, Springer Science & Business Media, 2013.
[38] Y. Kim, S.A. Choulis, J. Nelson, D.D. Bradley, S. Cook, J.R. Durrant, Device annealing effect in organic solar cells with blends of regioregular poly (3-hexylthiophene) and soluble fullerene, Appl. Phys. Lett. 86 (6) (2005) 3502.
[39] A.J. Moulé, K. Meerholz, Controlling morphology in polymer–fullerene mixtures, Adv. Mater. 20 (2) (2008) 240–245.
[40] Y. Liang, Z. Xu, J. Xia, S.-T. Tsai, Y. Wu, G. Li, C. Ray, L. Yu, For the bright future bulk heterojunction polymer solar cells with power conversion, Adv. Mater. 22 (2010).
[41] O. Wodo, B. Ganapathysubramanian, Modeling morphology evolution during solvent-based fabrication of organic solar cells, Comput. Mater. Sci. 55 (2012) 113–126.
[42] H.K. Kodali, B. Ganapathysubramanian, A computational framework to investigate charge transport in heterogeneous organic photovoltaic devices, Comput. Methods Appl. Mech. Eng. 247 (2012) 113–129.
[43] H.K. Kodali, B. Ganapathysubramanian, Sensitivity analysis of current generation in organic solar cells comparing bilayer, sawtooth, and bulk heterojunction morphologies, Solar Energy Mater. Solar Cells 111 (2013) 66–73.